QGIS Desktop 3.44 User Guide

QGIS Project

2026-03-14

Turinys

1 Pradžia **3**
- 1.1 QGIS diegimas . . . 3
  - 1.1.1 Diegimas iš dvejetainio kodo . . . 3
  - 1.1.2 Diegimas iš išeities kodo . . . 3
  - 1.1.3 Diegimas išoriniuose kaupikliuose . . . 4
  - 1.1.4 Pavyzdinių duomenų atsisiuntimas . . . 4
- 1.2 QGSI paleidimas ir išjungimas . . . 4
- 1.3 Pavyzdinė sesija: rastro ir vektorinių sluoksnių įkėlimas . . . 5

2 Darbas su projekto failais **13**
- 2.1 Įvadas į QGIS projektus . . . 13
- 2.2 Neteisingų kelių iki failų tvarkymas . . . 15
- 2.3 Išvesties kūrimas . . . 16

3 QGIS naudotojo sąsaja **17**
- 3.1 Meniu juosta . . . 18
  - 3.1.1 Projektas . . . 18
  - 3.1.2 Keisti . . . 19
  - 3.1.3 Vaizdas . . . 22
  - 3.1.4 Sluoksnis . . . 24
  - 3.1.5 Nustatymai . . . 27
  - 3.1.6 Priedai . . . 27
  - 3.1.7 Vektorius . . . 27
  - 3.1.8 Rastras . . . 29
  - 3.1.9 Duomenų bazės . . . 31
  - 3.1.10 Tinklas . . . 31
  - 3.1.11 Tinklelis . . . 31
  - 3.1.12 Apdorojimas . . . 32
  - 3.1.13 Pagalba . . . 32
  - 3.1.14 QGIS . . . 32
- 3.2 Skydeliai ir įrankinės . . . 33
  - 3.2.1 Įrankinės . . . 33
  - 3.2.2 Skydeliai . . . 34
- 3.3 Būsenos juosta . . . 34
  - 3.3.1 Vietos juosta . . . 34
  - 3.3.2 Veiksmų pranešimai . . . 35
  - 3.3.3 Žemėlapio drobės valdymas . . . 35
  - 3.3.4 Pranešimai . . . 36

4 Naršyklės skydelis **37**
- 4.1 Resursai, kurie gali būti atidaryti/paleisti naršyklėje . . . 39

4.2 Naršyklės skydelio aukščiausio lygio įrašai . . . 39
4.2.1 Parankiniai . . . 39
4.2.2 Erdvinės žymelės . . . 40
4.2.3 Projekto namų aplankas . . . 40
4.2.4 Diskai ir failų sistema . . . 40
4.2.5 Duomenų tipas . . . 41
4.2.6 Duomenų bazių įrašai . . . 41
4.2.7 Kaladėlės ir Žiniatinklio paslaugos . . . 45
4.2.8 Projektas ir palaikomi failų tipai . . . 46

5 QGIS konfigūracija **47**
5.1 Parinktys . . . 47
5.1.1 Bendrieji nustatymai . . . 48
5.1.2 Sistema . . . 50
5.1.3 Naudotojų profilių nustatymai . . . 52
5.1.4 CRS ir transformavimo nustatymai . . . 52
5.1.5 Duomenų šaltinių nustatymai . . . 55
5.1.6 Braižymo nustatymai . . . 59
5.1.7 Drobės ir legendos nustatymai . . . 62
5.1.8 Žemėlapio įrankių nustatymai . . . 64
5.1.9 Aukščio nustatymai . . . 68
5.1.10 3D Nustatymai . . . 68
5.1.11 Spalvų nustatymai . . . 69
5.1.12 Šriftų nustatymai . . . 70
5.1.13 Išdėstymų nustatymai . . . 71
5.1.14 Kintamųjų nustatymai . . . 72
5.1.15 Autentifikacijos nustatymai . . . 72
5.1.16 Tinklo nustatymai . . . 73
5.1.17 GPS nustatymai . . . 75
5.1.18 Atradėjo nustatymai . . . 77
5.1.19 Greitinimo nustatymai . . . 79
5.1.20 IDE nustatymai . . . 79
5.1.21 Apdorojimo nustatymai . . . 83
5.1.22 Sudėtingesni nustatymai . . . 83
5.2 Darbas su naudotojo profiliais . . . 84
5.2.1 Koncepcija . . . 84
5.2.2 Naudotojo profilio nustatymas . . . 85
5.3 Projekto savybės . . . 86
5.3.1 Bendros savybės . . . 86
5.3.2 Metaduomenų savybės . . . 89
5.3.3 Vaizdo nustatymai . . . 90
5.3.4 CRS savybės . . . 90
5.3.5 Transformacijų savybės . . . 91
5.3.6 Stilių savybės . . . 91
5.3.7 Spalvų savybės . . . 92
5.3.8 Duomenų šaltinių savybės . . . 93
5.3.9 Ryšių savybės . . . 95
5.3.10 Kintamųjų savybės . . . 96
5.3.11 Makro savybės . . . 96
5.3.12 QGIS serverio savybės . . . 97
5.3.13 Laiko savybės . . . 98
5.3.14 Aukščio savybės . . . 98
5.3.15 Jutiklių savybės . . . 101
5.4 Tinkinimas . . . 102
5.5 Klavišų kombinacijos . . . 104
5.6 QGIS paleidimas su sudėtingesniais nustatymais . . . 105
5.6.1 Komandinė eilutė ir aplinkos kintamieji . . . 105
5.6.2 QGIS diegimas organizacijoje . . . 111

6 Darbas su projekcijomis **113**
6.1 Projekcijų palaikymo apžvalga . . . 113
6.2 Sluoksnio koordinačių atskaitos sistemos . . . 113
6.3 Projekto koordinačių ataskaitos sistemos . . . 115
6.4 Koordinačių atskaitos sistemos parinkiklis . . . 117
6.5 Sava koordinačių atskaitos sistema . . . 118
6.5.1 NT-v2-transformacijos QGIS integracija . . . 119
6.6 Duomenų transformacija . . . 120

7 Žemėlapių vizualizavimas **123**
7.1 2D žemėlapio vaizdas . . . 123
7.1.1 Žemėlapio vaizdo tyrinėjimas . . . 124
7.1.2 Žemėlapio braižymo kontrolė . . . 125
7.1.3 Nuo laiko priklausantis žemėlapio drobės valdymas . . . 126
7.1.4 Aukščio valdymo skydelis . . . 128
7.1.5 Žemėlapio apimčių žymelės . . . 129
7.1.6 Žemėlapio dekoravimas . . . 131
7.1.7 Anotacijų įrankiai . . . 138
7.1.8 Matavimas . . . 143
7.1.9 Papildomų žemėlapio vaizdų kūrimas . . . 146
7.1.10 Žemėlapio vaizdo eksportavimas . . . 147
7.2 3D žemėlapio vaizdas . . . 149
7.2.1 Scenos konfigūravimas . . . 151
7.2.2 Navigacijos parinktys . . . 159
7.2.3 Animacijos kūrimas . . . 160
7.2.4 3D vektoriniai sluoksniai . . . 160
7.3 Aukščio profilio vaizdas . . . 161
7.3.1 Naudotojo sąsaja . . . 161
7.3.2 Aukščio profilio kūrimas . . . 163
7.3.3 Sąveika su profiliu vaizdais . . . 164
7.3.4 Aukščio profilio eksportavimas . . . 165

8 Bendri įrankiai **167**
8.1 Kontekstinė pagalba . . . 167
8.2 Skydeliai . . . 167
8.2.1 Sluoksnių skydelis . . . 167
8.2.2 Sluoksnio stiliaus skydelis . . . 175
8.2.3 Sluoksnių rikiuotės skydelis . . . 178
8.2.4 Apžvalgos skydelis . . . 178
8.2.5 Pranešimų žurnalo skydelis . . . 179
8.2.6 Atšaukimo/pakartojimo skydelis . . . 179
8.2.7 Statistinės suvestinės skydelis . . . 179
8.2.8 Tinkinimo/kūrimo įrankių skydelis . . . 180
8.3 Išorinių projektų sluoksnių įtraukimas . . . 184
8.4 Sąveika su geoobjektais . . . 185
8.4.1 Geoobjektų pažymėjimas . . . 185
8.4.2 Geoobjektų identifikavimas . . . 188
8.5 Sluoksnio savybių įrašymas ir bendrinimas . . . 192
8.5.1 Savo stilių tvarkymas . . . 192
8.5.2 Stilių saugojimas faile ar duomenų bazėje . . . 194
8.5.3 Sluoksnio apibrėžimo failas . . . 196
8.6 Jūsų duomenų dokumentavimas . . . 196
8.6.1 Metaduomenys . . . 196
8.6.2 Sluoksnio pastabos . . . 197
8.7 Reikšmių saugojimas kintamuosiuose . . . 198
8.8 Bendri valdikliai . . . 200
8.8.1 Spalvų parinkiklis . . . 200
8.8.2 Simbolių valdiklis . . . 204

8.8.3 Nutolusio arba įtraukto failo parinkiklis . . . 204
8.8.4 Nuo mastelio priklausomo matomumo parinkimas . . . 205
8.8.5 Erdvinės apimties parinkimas . . . 205
8.8.6 Šrifto parinkiklis . . . 206
8.8.7 Vienetų parinkiklis . . . 207
8.8.8 Skaičių formatavimas . . . 208
8.8.9 Suliejimo režimai . . . 209
8.8.10 Nuo duomenų priklausančių permušimų nustatymai . . . 211
8.8.11 Autentifikacija . . . 214

9 Pakelkite lygį su išraiškomis **217**
9.1 Išraiškos . . . 217
9.1.1 Išraiškų teksto kūrėjas . . . 217
9.1.2 Funkcijų redaktorius . . . 222
9.2 Funkcijų sąrašas . . . 225
9.2.1 Agregavimo funkcijos . . . 225
9.2.2 Array Functions . . . 235
9.2.3 Color Functions . . . 247
9.2.4 Conditional Functions . . . 256
9.2.5 Conversions Functions . . . 259
9.2.6 CRS Functions . . . 264
9.2.7 Custom Functions . . . 265
9.2.8 Date and Time Functions . . . 265
9.2.9 Fields and Values . . . 276
9.2.10 Files and Paths Functions . . . 276
9.2.11 Form Functions . . . 279
9.2.12 Fuzzy Matching Functions . . . 279
9.2.13 General Functions . . . 281
9.2.14 Geometry Functions . . . 283
9.2.15 Layout Functions . . . 357
9.2.16 Map Layers . . . 358
9.2.17 Maps Functions . . . 360
9.2.18 Mathematical Functions . . . 364
9.2.19 Meshes Functions . . . 372
9.2.20 Operators . . . 374
9.2.21 Processing Functions . . . 383
9.2.22 Rasters Functions . . . 383
9.2.23 Record and Attributes Functions . . . 385
9.2.24 Relations . . . 394
9.2.25 Sensors Functions . . . 394
9.2.26 String Functions . . . 394
9.2.27 User Expressions . . . 405
9.2.28 Variables . . . 405
9.2.29 Recent Functions . . . 409

10 Stilių biblioteka **411**
10.1 Stilių tvarkyklė . . . 411
10.1.1 Stilių tvarkyklės dialogas . . . 411
10.1.2 Spalvų rampos kūrimas . . . 417
10.1.3 Legendos elemento formos kūrimas . . . 419
10.2 Simbolių parinkiklis . . . 421
10.2.1 Simbolio sluoksnių medis . . . 422
10.2.2 Simbolio konfigūravimas . . . 423
10.3 Užrašo nustatymas . . . 441
10.3.1 Užrašo teksto formatavimas . . . 443
10.3.2 Sąveikos su užrašais konfigūravimas . . . 452
10.4 3D simbolių kūrimas . . . 462
10.4.1 Taškų sluoksniai . . . 463

10.4.2 Linijų sluoksniai . . . 464
10.4.3 Poligonų sluoksniai . . . 465
10.4.4 Tekstūros šešėliavimas . . . 467
10.4.5 Taikymo pavyzdys . . . 467

11 Duomenų šaltinių valdymas **469**
11.1 Duomenų atidarymas . . . 469
11.1.1 Naršyklės skydelis . . . 471
11.1.2 DB tvarkyklė . . . 474
11.1.3 Tiekėjų teikiami įkėlimo įrankiai . . . 475
11.1.4 Sluoksnio metaduomenų paieškos skydelis . . . 496
11.1.5 QGIS savi formatai . . . 497
11.1.6 QLR - QGIS sluoksnio apibrėžimo failas . . . 497
11.1.7 Prisijungimas prie žiniatinklio paslaugų . . . 498
11.2 Sluoksnių kūrimas . . . 506
11.2.1 Naujų vektorinių sluoksnių kūrimas . . . 507
11.2.2 Creating new layers from an existing layer . . . 516
11.2.3 Creating new DXF files . . . 521
11.2.4 Creating new layers from the clipboard . . . 522
11.2.5 Creating SQL Query Layers . . . 523
11.2.6 Creating virtual layers . . . 526
11.3 Geo-koordinavimas . . . 528
11.3.1 Rastro sluoksnio geo-koordinavimas . . . 529
11.3.2 Georeferencing vector layer . . . 535
11.3.3 Show and adapt layer properties . . . 536
11.3.4 Configure the georeferencer . . . 536
11.4 Exploring Data Formats and Fields . . . 537
11.4.1 Raster data . . . 537
11.4.2 Vector Data . . . 537
11.4.3 Layers crossing 180° longitude . . . 545

12 Working with Vector Data **547**
12.1 Vektorių savybių dialogas . . . 547
12.1.1 Informacinės savybės . . . 548
12.1.2 Source Properties . . . 548
12.1.3 Symbology Properties . . . 552
12.1.4 Labels Properties . . . 575
12.1.5 Diagrams Properties . . . 587
12.1.6 Masks Properties . . . 592
12.1.7 3D View Properties . . . 593
12.1.8 Fields Properties . . . 595
12.1.9 Attributes Form Properties . . . 596
12.1.10 Joins Properties . . . 608
12.1.11 Auxiliary Storage Properties . . . 608
12.1.12 Actions Properties . . . 616
12.1.13 Display Properties . . . 621
12.1.14 Rendering Properties . . . 624
12.1.15 Laiko savybės . . . 626
12.1.16 Kintamųjų savybės . . . 627
12.1.17 Aukščio savybės . . . 627
12.1.18 Metaduomenų savybės . . . 629
12.1.19 Dependencies Properties . . . 629
12.1.20 Legend Properties . . . 629
12.1.21 QGIS serverio savybės . . . 630
12.1.22 Digitizing Properties . . . 632
12.2 Darbas su atributų lentele . . . 634
12.2.1 Įvadas: erdvinės ir neerdvinės lentelės . . . 634
12.2.2 Introducing the attribute table interface . . . 635

12.2.3 Interacting with features in an attribute table . . . 640
12.2.4 Editing attribute values . . . 644
12.2.5 Exploring features attributes through the Identify Tool . . . 648
12.3 Redagavimas . . . 650
12.3.1 Setting the snapping tolerance and search radius . . . 651
12.3.2 Snapping and Digitizing Options . . . 651
12.3.3 Topological editing . . . 654
12.3.4 Digitizing an existing layer . . . 656
12.3.5 Advanced digitizing . . . 667
12.3.6 Shape digitizing . . . 676
12.3.7 The Advanced Digitizing panel . . . 679
12.3.8 The Processing in-place layer modifier . . . 687
12.4 Skirtingų sluoksnių duomenų jungimas ir redagavimas . . . 689
12.4.1 Dviejų sluoksnių geoobjektų sujungimas . . . 689
12.4.2 Setting relations between multiple layers . . . 691
12.4.3 Storing and fetching an external resource . . . 705

13 Working with Raster Data **709**
13.1 Rastro savybių dialogas . . . 709
13.1.1 Informacinės savybės . . . 710
13.1.2 Source Properties . . . 710
13.1.3 Symbology Properties . . . 711
13.1.4 Transparency Properties . . . 721
13.1.5 Labels Properties . . . 722
13.1.6 Histogram Properties . . . 724
13.1.7 Rendering Properties . . . 725
13.1.8 Laiko savybės . . . 726
13.1.9 Pyramids Properties . . . 727
13.1.10 Aukščio savybės . . . 728
13.1.11 Metaduomenų savybės . . . 729
13.1.12 Legend Properties . . . 729
13.1.13 Display Properties . . . 730
13.1.14 QGIS serverio savybės . . . 731
13.1.15 Attribute Tables . . . 732
13.1.16 Identify raster cells . . . 737
13.2 Raster Analysis . . . 737
13.2.1 Raster Calculator . . . 737

14 Darbas su tinklelio duomenimis **741**
14.1 Kas yra tinklelis? . . . 741
14.2 Supported formats . . . 743
14.3 Mesh Dataset Properties . . . 744
14.3.1 Informacinės savybės . . . 745
14.3.2 Source Properties . . . 746
14.3.3 Symbology Properties . . . 747
14.3.4 Labels Properties . . . 752
14.3.5 3D View Properties . . . 754
14.3.6 Rendering Properties . . . 756
14.3.7 Laiko savybės . . . 757
14.3.8 Aukščio savybės . . . 757
14.3.9 Metaduomenų savybės . . . 759
14.4 Exploring mesh datasets . . . 759
14.5 Editing a mesh layer . . . 760
14.5.1 Overview of the mesh digitizing tools . . . 761
14.5.2 Exploring the Z value assignment logic . . . 761
14.5.3 Selecting mesh elements . . . 763
14.5.4 Modifying mesh elements . . . 764
14.5.5 Reindexing meshes . . . 768

14.6 Mesh Calculator . . . 768

15 Working with Vector Tiles **771**
15.1 What are Vector Tiles? . . . 771
15.2 Supported Formats . . . 772
15.3 Vector Tiles Dataset Properties . . . 773
15.3.1 Informacinės savybės . . . 773
15.3.2 Source Properties . . . 773
15.3.3 Symbology and Labels Properties . . . 774
15.3.4 Rendering Properties . . . 775
15.3.5 Metaduomenų savybės . . . 776
15.4 QGIS Server Properties . . . 776

16 Darbas su taškų debesimis **777**
16.1 Įvadas į taškų debesis . . . 777
16.2 Point Clouds Properties . . . 778
16.2.1 Informacinės savybės . . . 778
16.2.2 Source Properties . . . 779
16.2.3 Symbology Properties . . . 782
16.2.4 3D View Properties . . . 790
16.2.5 Rendering Properties . . . 792
16.2.6 Aukščio savybės . . . 792
16.2.7 Metaduomenų savybės . . . 794
16.2.8 Statistics Properties . . . 794
16.3 Virtual point cloud . . . 795

17 Working with 3D Tiles **797**
17.1 What are 3D Tiles? . . . 797
17.1.1 State of 3D Tiles Support . . . 798
17.2 3D Tiles Properties . . . 799
17.2.1 Informacinės savybės . . . 799
17.2.2 Source Properties . . . 799
17.2.3 Symbology Properties . . . 799
17.2.4 3D View Properties . . . 800
17.2.5 Rendering Properties . . . 801
17.2.6 Aukščio savybės . . . 801
17.2.7 Metaduomenų savybės . . . 801

18 Laying out the maps **803**
18.1 Spausdinimo išdėstymo apžvalga . . . 803
18.1.1 Paprastos sesijos pradedantiesiems . . . 803
18.1.2 The Layout Manager . . . 804
18.1.3 Menus, tools and panels of the print layout . . . 805
18.2 Layout Items . . . 818
18.2.1 Layout Items Common Options . . . 818
18.2.2 The Map Item . . . 824
18.2.3 The 3D Map Item . . . 834
18.2.4 The Label Item . . . 835
18.2.5 The Legend Item . . . 840
18.2.6 The Scale Bar Item . . . 848
18.2.7 Lentelių elementai . . . 851
18.2.8 The Marker, Picture and North Arrow Items . . . 860
18.2.9 The Elevation Profile Item . . . 864
18.2.10 The HTML Frame Item . . . 866
18.2.11 The Shape Items . . . 868
18.3 Išvesties sukūrimas . . . 872
18.3.1 Eksporto nustatymai . . . 873
18.3.2 Export as Image . . . 873
18.3.3 Export as SVG . . . 874

18.3.4 Export as PDF . . . 875
18.3.5 Generate an Atlas . . . 877
18.4 Creating a Report . . . 883
18.4.1 What is it? . . . 883
18.4.2 Get started . . . 883
18.4.3 Layout Report Workspace . . . 884
18.4.4 Eksporto nustatymai . . . 899

19 Darbas su OGC / ISO protokolais 901
19.1 WMS/WMTS Client . . . 902
19.1.1 Overview of WMS Support . . . 902
19.1.2 Overview of WMTS Support . . . 902
19.1.3 Selecting WMS/WMTS Servers . . . 903
19.1.4 Loading WMS/WMTS Layers . . . 905
19.1.5 Kaladėlių rinkiniai . . . 907
19.1.6 Using the Identify Tool . . . 907
19.1.7 Viewing Properties . . . 908
19.1.8 Show WMS legend graphic in table of contents and layout . . . 912
19.2 WCS Client . . . 912
19.3 WFS and WFS-T Client . . . 914
19.4 SensorThings . . . 917
19.4.1 Setting connection . . . 917
19.4.2 Loading SensorThings data . . . 919
19.4.3 Working with a vector layer from SensorThings . . . 920
19.5 STAC (SpatioTemporal Asset Catalogs) . . . 920
19.5.1 Setting connection . . . 921
19.5.2 Browsing STAC Catalogs . . . 922
19.5.3 Filtering and Searching STAC Items . . . 923
19.6 ArcGIS REST Server . . . 925

20 Working with GPS Data 927
20.1 Introducing GNSS/GPS Data . . . 927
20.1.1 What is GPS? . . . 927
20.1.2 Transferring or loading GPS data . . . 928
20.2 Live GPS tracking . . . 930
20.2.1 GPS toolbar . . . 930
20.2.2 GPS Information Panel . . . 932
20.2.3 Connect to a Bluetooth GPS for live tracking . . . 934
20.2.4 Examples of GPS devices connection in QGIS . . . 934

21 Autentifikacijos sistema 937
21.1 Authentication System Overview . . . 937
21.1.1 Authentication database . . . 938
21.1.2 Custom authentication databases . . . 938
21.1.3 Master password . . . 939
21.1.4 Authentication Configurations . . . 940
21.1.5 Authentication Methods . . . 943
21.1.6 Master Password and Auth Config Utilities . . . 950
21.1.7 Using authentication configurations . . . 951
21.1.8 Python bindings . . . 951
21.2 User Authentication Workflows . . . 952
21.2.1 HTTP(S) authentication . . . 952
21.2.2 Database authentication . . . 953
21.2.3 PKI authentication . . . 954
21.2.4 Handling bad layers . . . 961
21.2.5 Changing authentication config ID . . . 961
21.2.6 QGIS Server support . . . 962
21.2.7 SSL server exceptions . . . 963

22 GRASS GIS Integration **967**
22.1 Demo dataset . . . 967
22.2 Loading GRASS raster and vector layers . . . 967
22.3 Importing data into a GRASS LOCATION via drag and drop . . . 968
22.4 Managing GRASS data in QGIS Browser . . . 968
22.5 GRASS Options . . . 968
22.6 Starting the GRASS plugin . . . 968
22.7 Opening GRASS mapset . . . 969
22.8 GRASS LOCATION and MAPSET . . . 969
22.9 Importing data into a GRASS LOCATION . . . 969
22.9.1 Creating a new GRASS LOCATION . . . 970
22.9.2 Adding a new MAPSET . . . 971
22.10 The GRASS vector data model . . . 972
22.11 Creating a new GRASS vector layer . . . 972
22.12 Digitizing and editing a GRASS vector layer . . . 973
22.13 The GRASS region tool . . . 975
22.14 The GRASS Toolbox . . . 975
22.14.1 Working with GRASS modules . . . 976
22.14.2 GRASS module examples . . . 979
22.14.3 Customizing the GRASS Toolbox . . . 981

23 QGIS processing framework **983**
23.1 Įvadas . . . 983
23.2 Apdorojimo karkaso konfigūravimas . . . 986
23.2.1 Bendra . . . 986
23.2.2 Menus . . . 988
23.2.3 Models and Scripts . . . 988
23.2.4 Providers . . . 988
23.3 Įrankinė . . . 989
23.3.1 The algorithm dialog . . . 991
23.3.2 Data objects generated by algorithms . . . 999
23.4 The history manager . . . 1000
23.4.1 The processing history . . . 1000
23.4.2 The processing log . . . 1001
23.5 Modelių dizaineris . . . 1001
23.5.1 Modelių dizainerio sąsaja . . . 1002
23.5.2 Creating a model . . . 1005
23.5.3 Saving and loading models . . . 1014
23.5.4 Editing a model . . . 1014
23.6 The batch processing interface . . . 1016
23.6.1 Įvadas . . . 1016
23.6.2 The parameters table . . . 1017
23.6.3 Filling the parameters table . . . 1018
23.6.4 Executing the batch process . . . 1019
23.7 Using processing algorithms from the console . . . 1019
23.7.1 Calling algorithms from the Python console . . . 1020
23.7.2 Creating scripts and running them from the toolbox . . . 1024
23.7.3 Pre- and post-execution script hooks . . . 1027
23.8 Apdorojimo naudojimas komandinėje eilutėje . . . 1027
23.9 Naujų apdorojimo algoritmų rašymas Python scenarijais . . . 1029
23.9.1 Extending QgsProcessingAlgorithm . . . 1029
23.9.2 The @alg decorator . . . 1033
23.9.3 Handing algorithm output . . . 1035
23.9.4 Communicating with the user . . . 1035
23.9.5 Documenting your scripts . . . 1036
23.9.6 Flags . . . 1036
23.9.7 Best practices for writing script algorithms . . . 1036
23.10 Apdorojimo algoritmų įvesties ir išvesties parametrai . . . 1036

23.10.1 Input types . . . 1036
23.10.2 Output types . . . 1039
23.11 Išorinių aplikacijų konfigūravimas . . . 1039
23.11.1 A note for Windows users . . . 1040
23.11.2 A note on file formats . . . 1040
23.11.3 A note on vector layer selections . . . 1040
23.11.4 Using third-party Providers . . . 1040

24 Processing providers and algorithms **1051**
24.1 QGIS algorithm provider . . . 1051
24.1.1 3D kaladėlės . . . 1051
24.1.2 Kartografija . . . 1053
24.1.3 Check Geometry . . . 1073
24.1.4 Duomenų bazės . . . 1119
24.1.5 Failų įrankiai . . . 1126
24.1.6 Fix Geometry . . . 1129
24.1.7 GPS . . . 1155
24.1.8 Interpoliacija . . . 1158
24.1.9 Sluoksnio įrankiai . . . 1172
24.1.10 Tinklelis . . . 1175
24.1.11 Metadata tools . . . 1188
24.1.12 Modeliuotojo įrankiai . . . 1193
24.1.13 Tinklo analizė . . . 1203
24.1.14 Diagramos . . . 1218
24.1.15 Taškų masyvo keitimas . . . 1231
24.1.16 Taškų debesies duomenų valdymas . . . 1238
24.1.17 Taškų masyvo ištraukimas . . . 1251
24.1.18 Rastro analizė . . . 1257
24.1.19 Rastro kūrimas . . . 1321
24.1.20 Rastro paviršiaus analizė . . . 1339
24.1.21 Rastro įrankiai . . . 1352
24.1.22 Vektorių analizė . . . 1364
24.1.23 Vektorių padengimas . . . 1389
24.1.24 Vektorių kūrimas . . . 1392
24.1.25 Bendri vektoriniai . . . 1421
24.1.26 Vektorinė geometrija . . . 1460
24.1.27 Vektorių persidengimai . . . 1581
24.1.28 Vektorių parinkimas . . . 1601
24.1.29 Vektorinė lentelė . . . 1619
24.1.30 Vektorinės kaladėlės . . . 1636
24.2 GDAL algorithm provider . . . 1640
24.2.1 Rastro analizė . . . 1640
24.2.2 Rastro keitimas . . . 1672
24.2.3 Rastro ištraukimas . . . 1680
24.2.4 Rastro įvairūs . . . 1689
24.2.5 Rastro projekcijos . . . 1710
24.2.6 Vektorių keitimas . . . 1715
24.2.7 Vektorių geoapdorojimas . . . 1722
24.2.8 Įvairūs vektoriniai . . . 1732

25 Priedai **1745**
25.1 QGIS priedai . . . 1745
25.1.1 Baziniai išoriniai priedai . . . 1745
25.1.2 The Plugins Dialog . . . 1746
25.2 Using QGIS Core Plugins . . . 1749
25.2.1 DB Manager Plugin . . . 1749
25.2.2 Geometry Checker Plugin . . . 1754
25.2.3 MetaSearch Catalog Client . . . 1758

25.2.4 Offline Editing Plugin . . . 1765
25.2.5 Topology Checker Plugin . . . 1767
25.3 QGIS Python konsolė . . . 1769
25.3.1 The Interactive Console . . . 1770
25.3.2 The Code Editor . . . 1772

26 Priedai **1775**
26.1 Appendix C: QGIS File Formats . . . 1775
26.1.1 QGS/QGZ - The QGIS Project File Format . . . 1775
26.1.2 QLR - The QGIS Layer Definition file . . . 1777
26.1.3 QML - The QGIS Style File Format . . . 1778
26.2 Appendix D: QGIS R script syntax . . . 1779
26.2.1 Inputs . . . 1780
26.2.2 Išvestys . . . 1780
26.2.3 Syntax Summary for QGIS R scripts . . . 1780
26.2.4 Pavyzdžiai . . . 1782
26.3 Priedas E: QGIS aplikacijos tinklo jungtys . . . 1784

27 Literature and Web References **1787**

SKYRIUS 1

Pradžia

Šiame skyriuje pateikiama trumpa apžvalga kaip diegti QGIS, atsisiųsti QGIS pavyzdinius duomenis bei vykdyti pirmą paprastą rastro ir vektorinių duomenų vizualizavimą.

1.1 QGIS diegimas

QGIS projektas teikia skirtingus QGIS diegimo būdus, priklausomai nuo jūsų platformos.

1.1.1 Diegimas iš dvejetainio kodo

Standartiniai diegikliai paruošti MS Windows ir macOS. Dvejetainiai paketai (rpm ir deb) arba programų repozitorijos teikiamos daugumai GNU/Linux distribucijų.

Daugiau informacijos ir instrukcijas jūsų operacinei sistemai rasite https://qgis.org/download/.

1.1.2 Diegimas iš išeities kodo

Jei jums reikia sukurti QGIS iš išeities kodo, prašome skaityti diegimo instrukcijas. Jos platinamos kartu su QGIS išeities kodu faile `INSTALL`. Jį taipogi galite rasti internete adresu https://github.com/qgis/QGIS/blob/release-3_44/INSTALL.md.

Jei norite sukurti konkrečią laidą, o ne šiuo metu kuriamą versiją, aukščiau minimoje nuorodoje turėtumėte pakeisti `master` su norimos laidos šaka (paprastai forma `release-X_Y` form) (gali skirtis diegimo instrukcijos).

1.1.3 Diegimas išoriniuose kaupikliuose

QGIS galima įdiegti flash kaupiklyje (su priedais ir nustatymais). Tai pasiekiama nurodant *–profiles-path* parinktį, kuri permuša numatytąjį *naudotojo profilio* kelią ir priverčia **QSettings** taipogi naudoti šį aplanką. Daugiau informacijos rasite skiltyje *Sistema*.

1.1.4 Pavyzdinių duomenų atsisiuntimas

Šiame naudotojų gide yra pavyzdžiai, kurie remiasi QGIS pavyzdiniais duomenimis (dar vadinamais `Aliaskos duomenų rinkiniu`). Atsisiųskite pavyzdinius duomenis iš https://github.com/qgis/QGIS-Sample-Data/archive/master.zip ir išpakuokite archyvą bet kurioje jums patogioje jūsų sistemos vietoje.

Aliaskos duomenų rinkinyje yra visi GIS duomenys, kurie naudojami pavyzdžiuose ir šio vadovo ekrano nuotraukose. Ten pat yra ir nedidelė GRASS duombazė. QGIS pavyzdinių duomenų projekcija yra Alaska Albers Equal Area su pėda kaip vienetu. EPSG kodas yra 2964.

```
PROJCS["Albers Equal Area",
GEOGCS["NAD27",
DATUM["North_American_Datum_1927",
SPHEROID["Clarke 1866",6378206.4,294.978698213898,
AUTHORITY["EPSG","7008"]],
TOWGS84[-3,142,183,0,0,0,0],
AUTHORITY["EPSG","6267"]],
PRIMEM["Greenwich",0,
AUTHORITY["EPSG","8901"]],
UNIT["degree",0.0174532925199433,
AUTHORITY["EPSG","9108"]],
AUTHORITY["EPSG","4267"]],
PROJECTION["Albers_Conic_Equal_Area"],
PARAMETER["standard_parallel_1",55],
PARAMETER["standard_parallel_2",65],
PARAMETER["latitude_of_center",50],
PARAMETER["longitude_of_center",-154],
PARAMETER["false_easting",0],
PARAMETER["false_northing",0],
UNIT["us_survey_feet",0.3048006096012192]]
```

Jei ketinate naudoti QGIS kaip grafinę sąsają GRASS, galite rasti kelis pavyzdinius duomenų rinkinius (pvz. Spearfish arba South Dakota) oficialioje GRASS GIS svetainėje https://grass.osgeo.org/download/data/.

1.2 QGSI paleidimas ir išjungimas

QGIS galima paleisti kaip bet kurią kitą jūsų kompiuterio programą. Tai reiškia, kad jūs galite QGIS paleisti:

- naudodami Aplikacijų meniu, Start meniu arba Dock
- du kartus paspaudę ant Aplikacijos aplanko ar darbastalio piktogramos
- du kartus paspaudę ant esamo QGIS projekto failo (su `.qgz` ar `.qgs` praplėtimu). Pastebėtina, kad tai atvers projektą.
- parašę `qgis` komandinėje eilutėje (darant prielaidą, kad QGIS pridėtas į jūsų PATH arba jūs esate diegimo aplanke)

QGIS galite išjungti:

- meniu parinktimi option *Projektas* ► *Išjungti QGIS* arba naudokite klavišų kombinaciją `Ctrl+Q`
- *QGIS* ► *Išjungti QGIS* arba naudokite klavišų kombinaciją `Cmd+Q`
- arba naudokite raudoną kryžiuką viršutiniame dešiniame pagrindinio aplikacijos lango kampe.

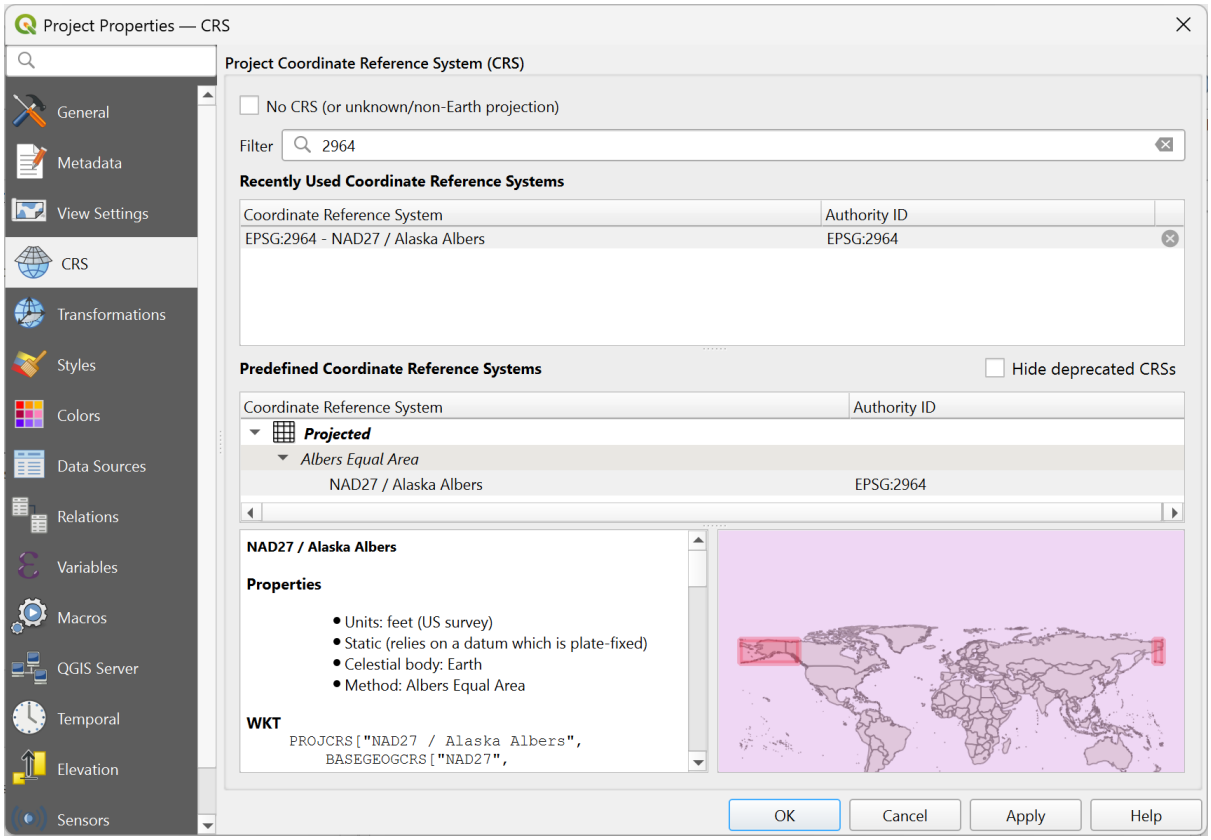
1.3 Pavyzdinė sesija: rastro ir vektorinių sluoksnių įkėlimas

Dabar jūs *įdiegėte QGIS* ir *pavyzdinį duomenų rinkinį*, mes parodysime pirmą pavyzdinę sesiją. Šiame pavyzdyje vizualizuosime rastro ir vektorinį sluoksnį. Mes naudosime:

- `landcover` rastro sluoksnį (`qgis_sample_data/raster/landcover.img`)
- ir `lakes` vektorinį sluoksnį (`qgis_sample_data/gml/lakes.gml`)

Kur `qgis_sample_data` yra kelias iki išpakuoto duomenų rinkinio.

1. Paleiskite QGIS kaip nurodyta *QGSI paleidimas ir išjungimas*.
2. Duomenys, su kuriais dirbsime, naudoja `Albers Equal Area` projekciją, todėl atitinkamai nustatykime projekto CRS:
    1. QGIS naudotojo sąsajos apačioje dešinėje spauskite mygtuką Parinkti projekciją. Atsidaro projekto savybių dialogas su aktyvia kortele *CRS*.
    2. Lauke *Filtras* įrašykite `2964`.
    3. Parinkite eilutę su CRS pavadinimu `NAD27 / Alaska Albers`.



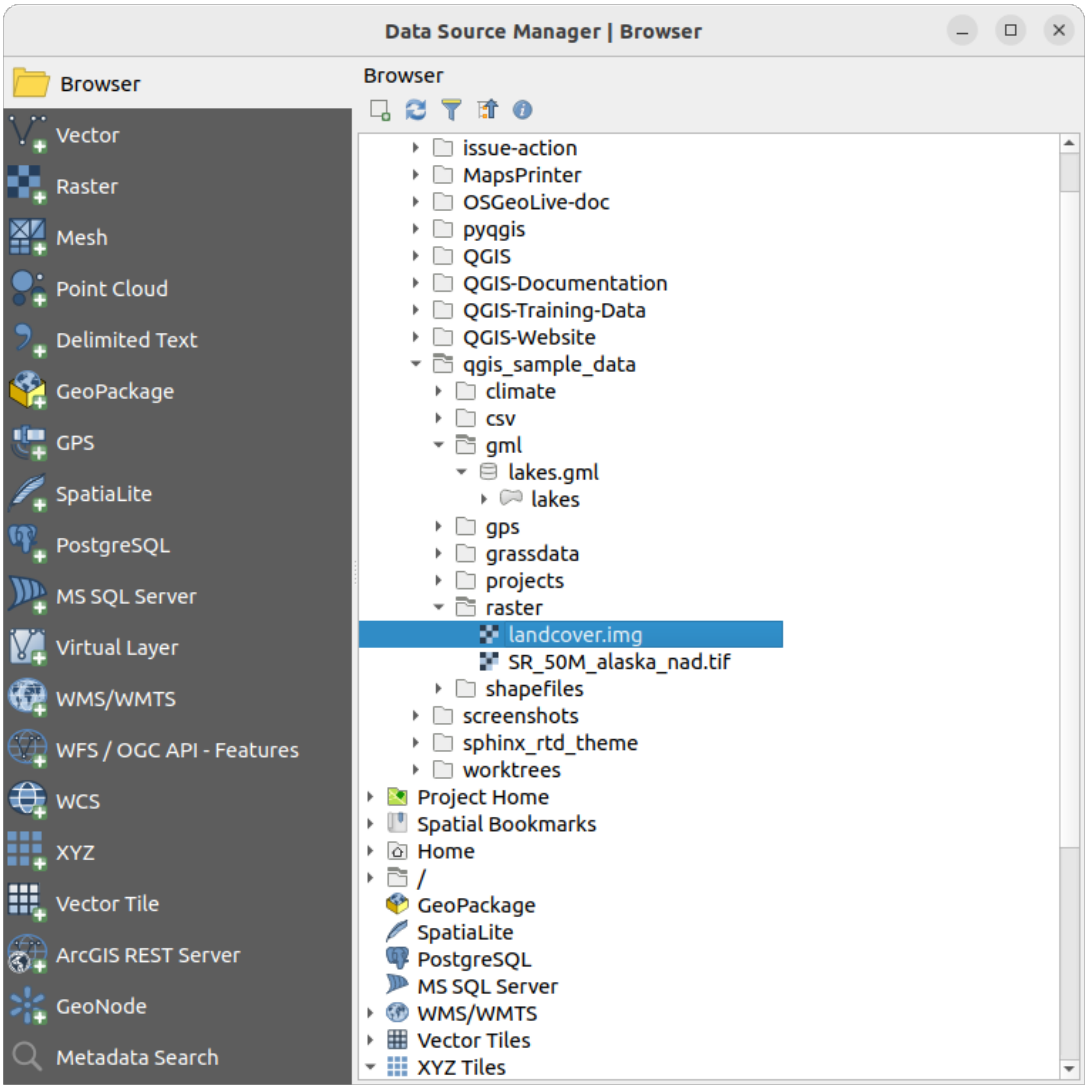
1.1 Fig. : Parinkite duomenų koordinačių atskaitos sistemą

4. Spauskite *Gerai*

> **Pastaba**
>
> Kol kas galite ignoruoti/uždaryti pranešimą „ballpark transform“, kuris gali pasirodyti.

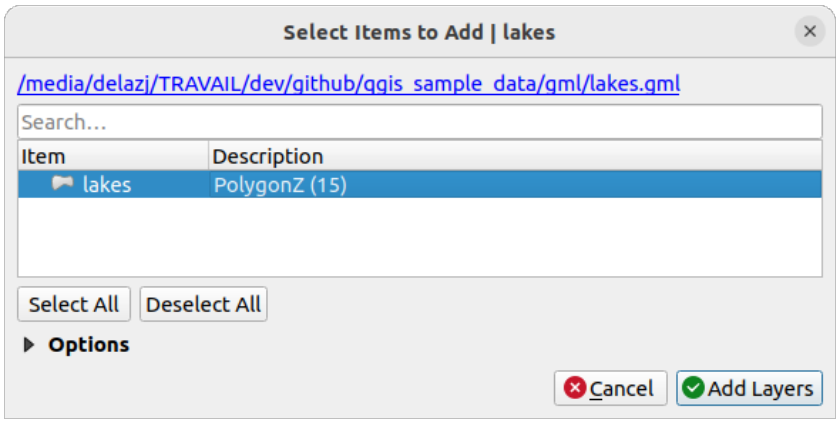
3. Įkelkite failus į QGIS:

1. Spauskite Duomenų šaltinių tvarkyklės piktogramą. Duomenų šaltinių tvarkyklė turėtų atsidaryti Naršymo režime.
2. Suraskite aplanką `qgis_sample_data/raster/`
3. Parinkite ERDAS IMG failą *landcover.img* ir du kartus jį paspauskite. Žemės dangos sluoksnis pridedamas duomenų fone, o Duomenų šaltinių tvarkymo langas lieka atidarytas.



1.2 Fig. : Duomenų pridėjimas į naują QGIS projektą

4. Kad įkeltumėte ežerų duomenis, suraskite aplanką `qgis_sample_data/gml/` ir atitempkite *lakes.gml* failą virš QGIS pagrindinio dialogo. (Arba tiesiog spauskite, kaip minėta, du kartus.)
5. Atsidaro dialogas *Parinkite elementus pridėjimui* ir nuskenuojamas failas. Taip yra todėl, kad `.gml` formatas leidžia saugoti daugiau nei vieną sluoksnį viename faile.



1.3 Fig. : Parinkite failo sluoksnius

6. Mūsų atveju yra tik vienas sluoksnis *lakes*. Parinkite jį ir spauskite *Pridėti sluoksnius*.
7. Sluoksnis pridedamas į skydelį *Sluoksniai*

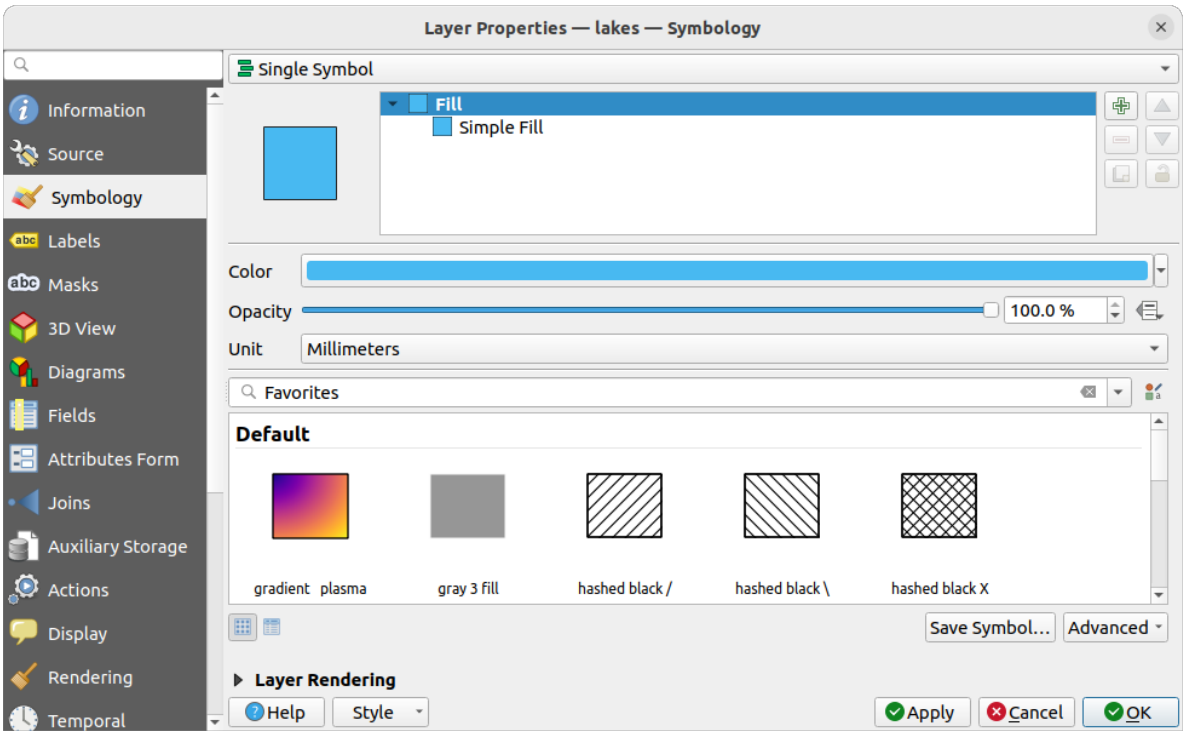
4. Uždarykite duomenų šaltinių tvarkyklės langą

Skydelyje *Sluoksniai* jūs galite pastebėti, kad prie sluoksnio *lakes* rodoma Sluoksnis neturi nustatytos koordinačių atskaitos sistemos. Pataisykime tai.

1. Spauskite ženkliuką . Atidaromas dialogas *Koordinačių atskaitos sistemos parinkimas*.
2. Kaip ir anksčiau, raskite ir pažymėkite CRS įrašą *NAD27 / Alaska Albers*.
3. Sauskite *Gerai*

Dabar jūs projekte turite du sluoksnius, nuspalvintus atsitiktinėmis spalvomis. Patinkinkime ežerų sluoksnį.

1. Parinkite Didinti įrankį *Navigacijos* įrankinėje
2. Priartinkite plotą su ežerais
3. Du kartus spauskite ant `lakes` sluoksnio žemėlapio legendoje, kad atvertumėte *Savybių* dialogą
4. Kad pakeistumėte ežerų spalvą:
    1. Spauskite ant *Simbologijos* kortelės
    2. Parinkite mėlyną pildymo spalvą

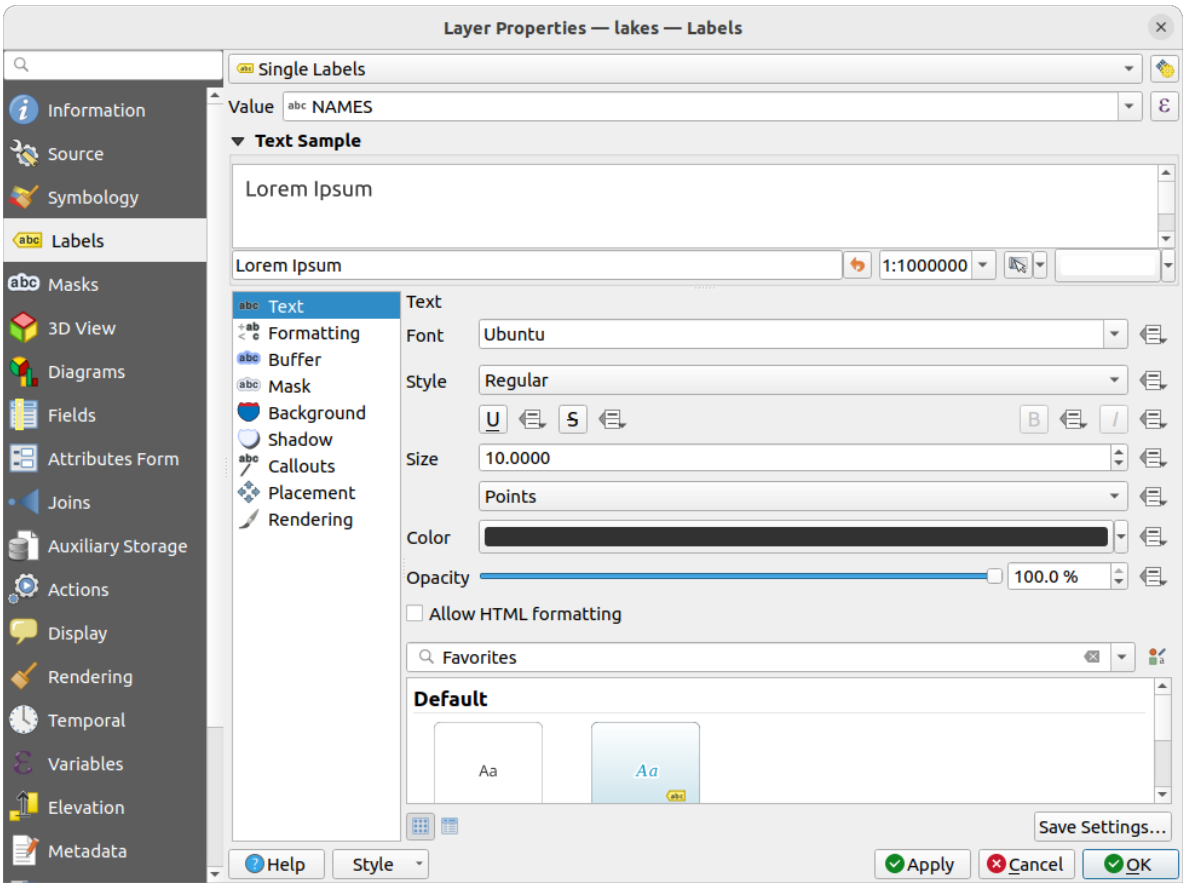


1.4 Fig. : Ežerų spalvos parinkimas

3. Spauskite *Gerai*. Ežerai dabar žemėlapio drobėje rodomi mėlynai.

5. Kad parodytumėte ežerų pavadinimus:

   1. Vėl atidarykite `lakes` sluoksnio *Savybių* dialogą
   2. Spauskite ant *Užrašų* kortelės
   3. Iškrentančiame sąraše parinkite *Atskiri užrašai*, kad įjungtumėte užrašus.
   4. *Reikšmė* sąraše parinkite lauką `NAMES`.

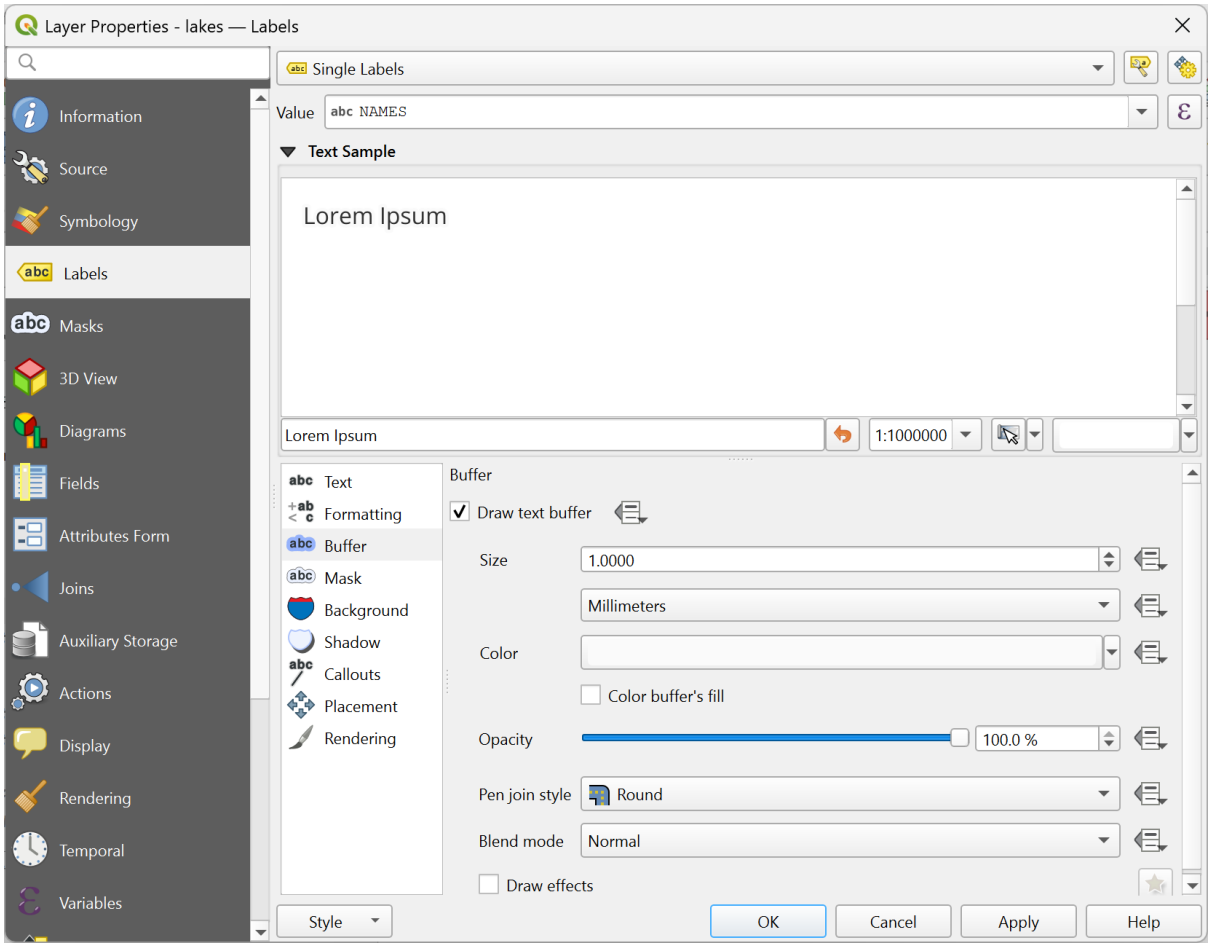


1.5 Fig. : Ežerų pavadinimų rodymas

5. Spauskite *Taikyti*. Pavadinimai dabar pasirodys virš ribų.

6. Galite pagerinti etikečių įskaitomumą pridėdami aplink baltą buferį:

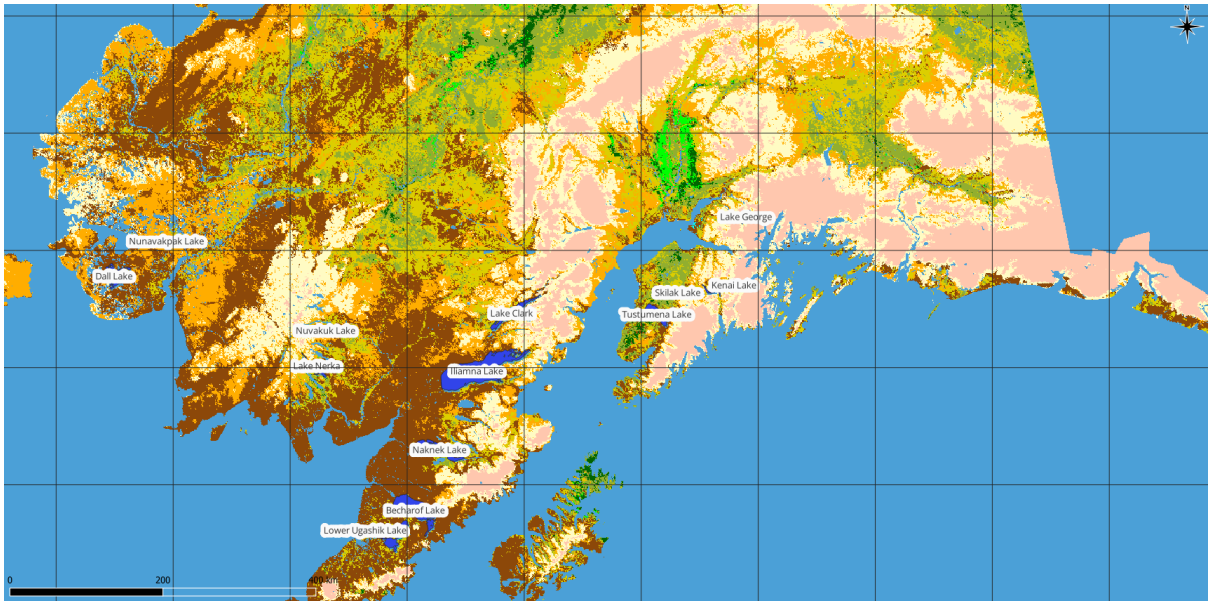
   1. Spauskite kortelę *Buferis* sąraše kairėje
   2. Įjunkite *Paišyti teksto buferį*
   3. Parinkite `3`, kaip buferio dydį
   4. Spauskite *Taikyti*
   5. Pažiūrėkite, ar rezultatas gerai atrodo, jei reikia, pakeiskite reikšmę.
   6. Tada spauskite *Gerai*, kad uždarytumėte *Sluoksnio savybių* dialogą ir pritaikytumėte pakeitimus.



1.6 Fig. : Buferių rodymas aplink užrašus

Dabar pridėkite kelias dekoracijas, kad apipavidalintumėte žemėlapį ir išeksportuotumėte jį iš QGIS:

1. Parinkite meniu *Rodymas* ► *Dekoracijos* ► *Mastelio juostą*
2. Atsivėrusiame dialoge įjunkite parinktį *Įjungti mastelio juostą*
3. Nustatykite parinktis taip, kaip norite
4. Spauskite *Gerai*
5. Analogiškai iš dekoracijų meniu pridėkite daugiau elementų (šiaurės rodyklę, autorių teisių etiketę…) į žemėlapio drobę su savom savybėm.
6. Spauskite *Projektas* ► *Importuoti/Eksportuoti* ► *Eksportuoti žemėlapį į pietinį…*
7. Atsidariusiame dialoge spauskite *Įrašyti*
8. Parinkite failo vietą, formatą ir patvirtinkite vėl spausdami *Įrašyti*.
9. Spauskite *Projektas* ► *Įrašyti…*, kad įrašytumėte jūsų pakeitimus kaip `.qgz` projekto failą.



1.7 Fig. : Su dekoracijomis eksportuoto žemėlapio rodymas

Viskas! Matote kaip paprasta QGIS vizualizuoti rastro ir vektorinius sluoksnius, konfigūruoti juos ir sukurti jūsų žemėlapį piešinio formatu, kurį galite naudoti kitose programose. Eikime toliau ir išmokime daugiau apie funkcionalumą, savybes ir nustatymus bei kaip juos naudoti.

Pastaba

Norėdami tęsti mokytis QGIS pažingsniui, tęskite Mokymo medžiagą.

SKYRIUS 2

Darbas su projekto failais

2.1 Įvadas į QGIS projektus

Jūsų QGIS sesijos būsena vadinasi projektu. QGIS vienu metu dirba su vienu projektu. Nustatymai gali būti arba projekto, arba aplikacijos lygmenyje naujiems projektams (daugiau informacijos skiltyje *Parinktys*). QGIS gali įrašyti jūsų darbo erdvės būseną į *QGIS projekto failą* naudojant meniu parinktis *Projektas* ► *Įrašyti* arba *Projektas* ► *Įrašyti kaip…*.

Pastaba

Jei projektas buvo pakeistas, antraštėje pasirodys simbolis * ir QGIS pagal nutylėjimą paklaus, ar norite įrašyti pakeitimus. Šią elgseną valdo nustatymas *Prireikus pasiūlyti įrašyti projekto ir duomenų šaltinio pakeitimus*, kurį rasite *Nustatymai* ► *Parinktys* ► *Bendra*.

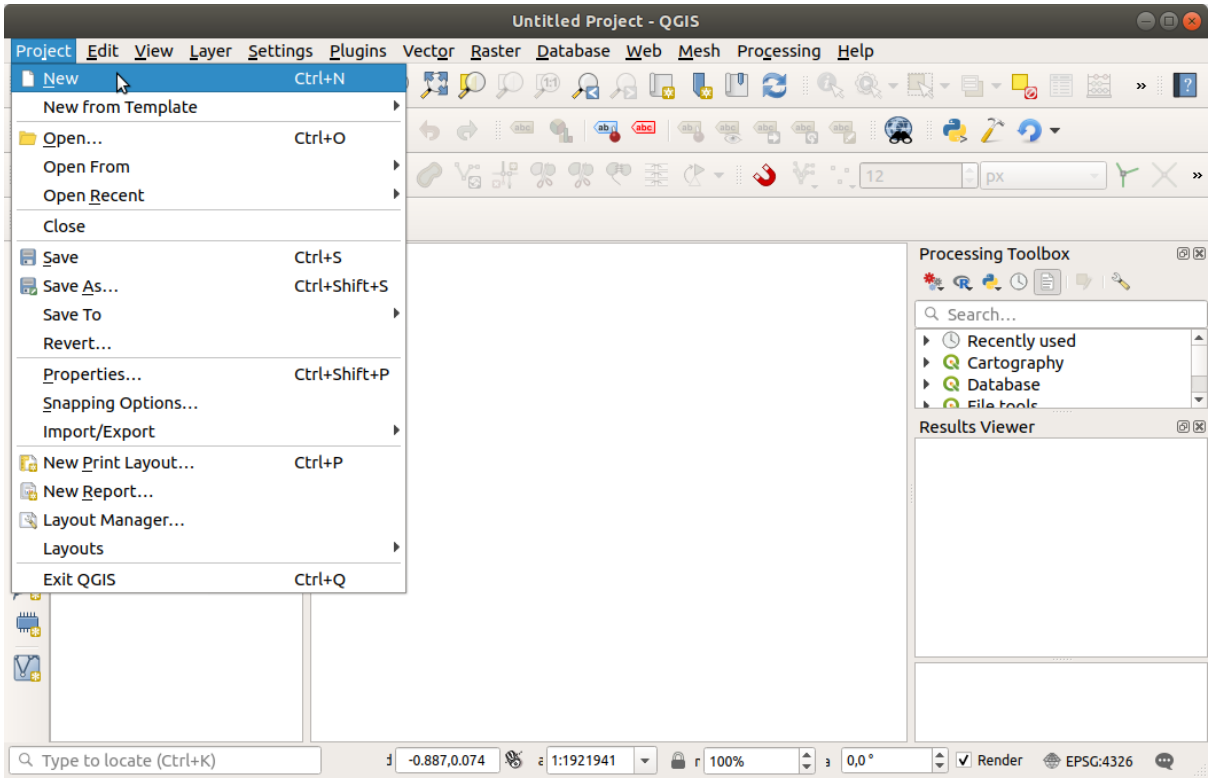
Galite įkelti esamą projektą naudodami *Projektas* ► *Atverti…*, *Projektas* ► *Naujas pagal šabloną* arba *Projektas* ► *Atverti ankstesnius* ►.

Iš *Naršyklės skydelio* jūs galite *Atverti projektą*, *Ištraukti simbolius…* – atveria stilių tvarkyklę simbolių eksportavimui į XML, jų pridėjimui į numatytąjį stilių ar eksportavimui kaip PNG/SVG. Jūs taipogi galite pasiekti dialogą *Failo savybės…*. Projekto failą galima išskleisti, kad pamatytumėte jo sluoksnius. Sluoksnio kontekstinis meniu siūlo kai kuriuos veiksmus kaip ir kitur naršyklėje.

Pradžioje rodomas *Projekto šablonų* ir *Paskutinių projektų* sąrašas su nuotraukomis, pavadinimais ir keliu (iki dešimties projektų). *Paskutiniai projektai* patogūs norint pasiekti paskutinius naudotus projektus. Du kartus spauskite ant įrašo, kad atvertumėte projektą ar projekto šabloną. Spauskite ant įrašo dešinį mygtuką, jei norite *Prisegti prie sąrašo*, *Atverti aplanką…* ar *Išimti iš sąrašo*. Jei matote savo projektą *Paskutinių projektų* sąraše bet nerandate jo savo failų tvarkyklėje, naudokite parinktį *Atverti aplanką…*, kuris padės jums rasti projektus, kuriuos galbūt pametėte, perkėlėte arba pervadinote. Taipogi galite *Išvalyti sąrašą*, jei norite išimti visus projektus iš *Paskutinių projektų* sąrašo. Jei turite prisegtų projektų, po veiksmo *Išvalyti sąrašą* bus parodytas klausimas, ar norite išimti ir prisegtus projektus. Taipogi galite pridėti sluoksnį, kad automatiškai sukurtumėte naują projektą. Sąrašas tada dings, jo vietą užims žemėlapio drobė.

Jei norite išvalyti jūsų sesiją ir pradėti iš naujo, parinkite *Projektas* ► *Naujas*. Jūsų bus paklausta, ar norite įrašyti dabartinį projektą, jei po jo paskutinio įrašymo buvo padaryti pakeitimai.

Kai atidarote naują projektą, antraštėje bus parašyta „Nepavadintas projektas“ tol, kol jūs jo neįrašysite.



2.1 Fig. : Naujo projekto kūrimas QGIS’e

Projekto faile įrašoma informacija apie:

- Pridėtus sluoksnius
- Kuriuose sluoksniuose galima daryti užklausas
- Sluoksnių savybės, įskaitant simbolizavimą ir stilius
- Sluoksnio pastabos
- 2D ir 3D žemėlapio vaizdai
- Projekcija ir kiekvieno žemėlapio vaizdas
- Paskutinė žiūrėta kiekvieno žemėlapio apimtis
- Spausdinimo išdėstymai
- Spausdinimo išdėstymo elementai su nustatymais
- Spausdinimo išdėstymo atlaso nustatymai
- Skaitmeninimo nustatymai
- Lentelių ryšiai
- Projekto makro
- Projekto numatytieji stiliai
- Priedų nustatymai
- QGIS serverio nustatymai iš OWS nustatymų kortelės Projekto savybėse
- DB tvarkyklėje įrašytos užklausos

Projekto failas įrašomas XML formatu (žiūrėkite *QGS/QGZ - The QGIS Project File Format*). Tai reiškia, kad failą galima redaguoti ir be QGIS, jei žinote, ką darote. Projekto failo formatas buvo kelis kartus atnaujintas. Senesnių QGIS versijų projektų failai gali nebeveikti tinkamai.

Pastaba

Pagal nutylėjimą QGIS įspės apie versijų skirtumus. Šią elgseną valdo meniu punkto *Nustatymai* ► *Parinktys* kortelės *Bendra* parinktis *Įspėti, kai atidaromas projekto failas, sukurtas senesne QGIS versija*.

Kai įrašote `.qgs` projekto failą, QGIS sukuria atsarginę failo kopiją tame pačiame aplanke, kaip ir projekto failas su praplėtimu `.qgs~`.

QGIS projektų plėtinys yra `.qgs`, bet įrašant iš QGIS, numatytas įrašymo būdas yra suspaustas formatas su plėtiniu `.qgz`. `.qgs` failas yra `.qgz` failo (zip archyvo) viduje, kartu su susijusia SQLite duombaze (`.qgd`) *papildomiems duomenims*. Šiuos failus galite pasiekti išpakavę `.qgz` failą.

Pastaba

Auxiliary Storage Properties mechanizmas daro suarchyvuotą projektą ypač naudingu, kadangi viduje laikomi ir papildomi duomenys.

Projektus taipogi galima įrašyti/įkelti į/iš PostgreSQL, GeoPackage ar Oracle duomenų bazės naudojant šiuos Projekto meniu elementus:

- *Projektas* ► *Atverti iš*
- *Proktas* ► *Įrašyti į*

Abu meniu elementai turi submeniu su keliomis projekto saugojimo parinktimis (PostgreSQL, GeoPackage ir Oracle). Paspaudus veiksmą atsidarys dialogas, kuriame galima bus parinkti GeoPackage jungtį ir projektą, PostgreSQL jungtį, schemą ir projektą, arba Oracle jungtį, savininką ir projektą.

Geopackage, PostgreSQL ar Oracle įrašytą projektą galima įkelti per QGIS naršyklės skydelį, arba du kartus ant jo paspaudus, arba pertempus į žemėlapio drobę.

2.2 Neteisingų kelių iki failų tvarkymas

Atidarant projektą QGIS gali nepavykti rasti duomenų šaltinius dėl to, kad neprieinamos paslaugos/duomenų bazės arba failai pervadinti ar perkelti į kitą vietą. QGIS tada atveria dialogą *Tvarkyti nepasiekiamus sluoksnius*, kuriame surašyti nerasti sluoksniai. Jūs galite:

- Du kartus spausti lauke *Duomenų šaltinis*, pataisyti kiekvieno sluoksnio kelią ir spausti *Pritaikyti pakeitimus*;
- Pažymėti eilutę, spausti *Naršyti*, nurodyti teisingą vietą ir spausti *Pritaikyti pakeitimus*;
- Spausti *Rasti automatiškai*, kad būtų naršoma po aplankus ir bandoma automatiškai rasti pataisymą visiems sugadintiems keliams. Turėkite omenyje, kad naršymas gali užtrukti daug laiko. Tada spauskite *Pritaikyti pakeitimus*.
- Ignoruoti pranešimą ir atidaryti jūsų projektą be sugadintų kelių spausdami *Palikti neprieinamus sluoksnius*. Tokiu atveju jūsų sluoksnis bus rodomas *Sluoksnių* skydelyje, bet be jokių duomenų, tol, kol jūs pataisysite kelią Neprieinamas sluoksnis! piktogramoje, kuri bus rodoma greta *Sluoksnių* skydelyje arba *Taisyti duomenų šaltinį…* sluoksnio kontekstiniame meniu.

  Po to, kai sluoksnio kelias pataisomas, įrankis *Taisyti duomenų šaltinį…* skenuoja kitus sugadintus kelius ir bando automatiškai pataisyti juos naudojant tokį patį kelią.

- *Išimti neprieinamus sluoksnius* iš projekto.

Paleidus QGIS iš komandinės eilutės naudojant parinktį *--skipbadlayers* padės jums praleisti pradžioje rodomą dialogą *Tvarkyti neprieinamus sluoksnius*.

2.3 Išvesties kūrimas

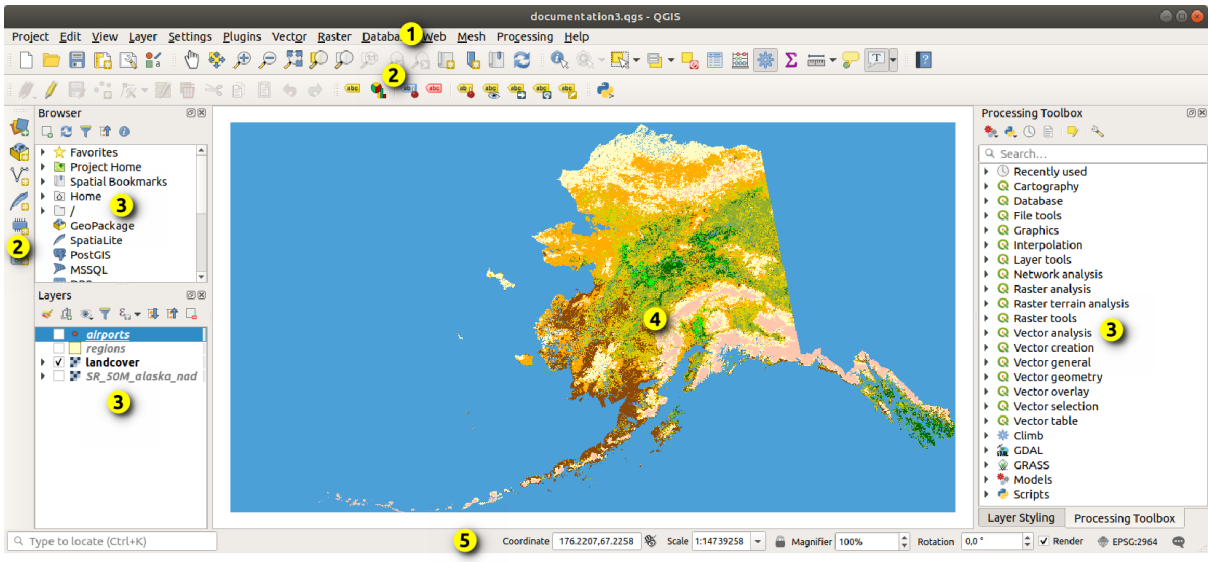
Yra keli būdai, kaip galima kurti QGIS sesijos išvestį. Mes jau minėjome įrašymą kaip projekto failą skyriuje *Įvadas į QGIS projektus*. Kiti būdai sukurti išvesti yra:

- Kurti piešinius: *Projektas* ► *Importuoti/Eksportuoti* ► *Eksportuoti žemėlapį į piešinį…* įrašo žemėlapio drobę kaip piešinį (PNG, JPG, TIFF…) nurodytu masteliu, rezoliucija ir dydžiu. Galima piešiniui įrašyti geopririšimo informaciją, tiesiog įjunkite varnelę *Pridėti geopririšimo informaciją (pridedama per pasaulio failą)*. Daugiau informacijos rasite *Žemėlapio vaizdo eksportavimas*.
- Eksportuoti į PDF failus: *Projektas* ► *Importuoti/Eksportuoti* ► *Eksportuoti žemėlapį į PDF…* išveda žemėlapio drobę į PDF nurodytu masteliu, rezoliucija su keliais išmanesniais nustatymais (paprastinimas, geopririšimas, …). Daugiau informacijos rasite *Žemėlapio vaizdo eksportavimas*.
- Eksportuoti į DXF failus: *Projektas* ► *Importuoti/Eksportuoti* ► *Eksportuoti projektą į DXF…* atveria dialogą, kur jūs galite nurodyti „Simbologijos režimą“, „Simbologijos mastelį“ ir vektorinius sluoksnius, kuriuos norite eksportuoti į DXF. Per „Simbologijos režimą“, simboliai iš pradinės QGIS simbologijos gali būti eksportuoti aukštos kokybės režimu (daugiau informacijos rasite skyriuje *Creating new DXF files*).
- Kurti žemėlapius: *Projektas* ► *Naujas spausdinimo išdėstymas…* atidaro dialogą, kur galite išdėstyti ir spausdinti dabartinę žemėlapio drobę (daugiau informacijos skyriuje *Laying out the maps*).

SKYRIUS 3

QGIS naudotojo sąsaja

QGIS grafinė naudotojo sąsaja (GUI) vaizduojama iliustracijoje žemiau (geltoni rutuliukai su skaičiais nuo 1 iki 5 rodo svarbius QGIS GUI elementus ir yra aptariami žemiau).



3.1 Fig. : QGIS GUI su Aliaskos pavyzdiniais duomenimis

Pastaba

Jūsų langų dekoracijos (antraštės juosta ir pan.) gali skirtis priklausomai nuo jūsų operacinės sistemos ir langų tvarkyklės.

Pagrindinis QGIS GUI (Fig. 3.1) susideda iš penkių komponentų / komponentų tipų:

1. *Meniu juostos*
2. *Įrankinių*
3. *Skydelių*

4. *Žemėlapio vaizdo*
5. *Būsenos juostos*

Žemiau pateikiami detalesni šių elementų aprašymai.

3.1 Meniu juosta

Meniu juosta suteikia prieigą prie QGIS funkcijų standartinio hierarchinio meniu pagalba. Žemiau aprašomi meniu, jų parinktys, susijusios piktogramos ir klavišų kombinacijos. Klavišų kombinacijas galima keisti (*Nustatymai* ► *Klavišų kombinacijos*).

Dauguma meniu parinkčių turi atitinkamus įrankius ir atvirkščiai. Kita vertus, meniu nėra organizuotas taip pat, kaip įrankinės. Meniu parinkčių vietos įrankinėse indikuojamos lentelėje žemiau. Priedai gali pridėti į meniu papildomas parinktis. Daugiau informacijos apie įrankius ir įrankines rasite skyriuje *Įrankinės*.

Pastaba

QGIS yra įvairioms platformoms skirta aplikacija. Įrankiai paprastai prieinami visose platformose, bet jie gali būti padėti skirtinguose meniu, priklausomai nuo operacinės sistemos. Sąraše žemiau rasite dažniausias vietas, įskaitant ir žinomas variacijas.

3.1.1 Projektas

Meniu *Projektas* teikia *projekto failų* prieigos ir išėjimo taškus. Jis teikia įrankius, skirtus:

- Kurti *Naują* projekto failą nuo nulio arba naudoti kitą projekto failą kaip šabloną (daugiau apie šablonų konfigūravimą žiūrėkite *Projekto failų parinktys*)
- *Atverti…* projektą iš failo, a GeoPackage, PostgreSQL ar Oracle duombazės
- *Uždaryti* projektą ar atstatyti jo paskutinę įrašytą būseną
- *Įrašyti* projektą į `.qgs` ar `.qgz` formato failą, arba kaip atskirą failą, arba į GeoPackage, PostgreSQL ar Oracle duomenų bazę
- Eksportuoti žemėlapio drobę į skirtingus formatus arba naudoti *spausdinimo išdėstymą*, jei reikia sudėtingesnės išvesties
- Nustatyti projekto savybes ar parinktis, kaip bus pritraukiamos geometrijos redaguojant

3.1 Table : Projekto meniu elementai

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Naujas	`Ctrl+N`	*Projekto*	*Įvadas į QGIS projektus*
Naujas pagal šabloną ►			*Įvadas į QGIS projektus*
Atverti…	`Ctrl+O`	*Projekto*	*Įvadas į QGIS projektus*
Atverti iš ►			*Įvadas į QGIS projektus*
► *GeoPackage…*			*Įvadas į QGIS projektus*
► *PostgreSQL…*			*Įvadas į QGIS projektus*
► *Oracle…*			*Įvadas į QGIS projektus*
Atverti ankstesnius ►	`Alt+J + R`		*Įvadas į QGIS projektus*
► *Valyti sąrašą*			*Įvadas į QGIS projektus*
Uždaryti			*Įvadas į QGIS projektus*
Įrašyti	`Ctrl+S`	*Projekto*	*Įvadas į QGIS projektus*

continues on next page

Table 3.1 – tęsinys iš praeito puslapio

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Įrašyti kaip…	Ctrl+Shift+S	*Projekto*	*Įvadas į QGIS projektus*
Įrašyti į ►			*Įvadas į QGIS projektus*
► Šablonus…			*Įvadas į QGIS projektus*
► GeoPackage…			*Įvadas į QGIS projektus*
► PostgreSQL…			*Įvadas į QGIS projektus*
Atstatyti…			
Savybės…	Ctrl+Shift+P		*Projekto savybės*
Pritraukimo parinktys…			*Snapping and Digitizing Options*
Importuoti/Eksportuoti ►			
► Eksportuoti žemėlapį į piešinį…			*Žemėlapio vaizdo eksportavimas*
► Eksportuoti žemėlapį į PDF…			*Žemėlapio vaizdo eksportavimas*
► Eksportuoti žemėlapį į DXF…			*Creating new DXF files*
► Importuoti sluoksnius iš DWG/DXF…			*DXF ar DWG failo importavimas*
Naujas spausdinimo išdėstymas…	Ctrl+P	*Projekto*	*Laying out the maps*
Nauja ataskaita…			*Creating a Report*
Maketų tvarkyklė…		*Projekto*	*Laying out the maps*
Maketai ►			*Laying out the maps*
Modeliai ►			*Modelių dizaineris*
Išjungti QGIS	Ctrl+Q		

macOS sistemoje, *Išjungti QGIS* komandą atitinka *QGIS ► Išjungti QGIS* (Cmd+Q).

3.1.2 Keisti

Meniu *Keisti* teikia daugumą įrankių sluoksnio atributų ar geometrijos redagavimui. Norėdami įjungti *Keisti* meniu parinktis, jūs turite persijungti į redagavimo režimą spausdami Perjungti redagavimą (daugiau informacijos rasite skyriuje *Redagavimas*).

3.2 Table : Keitimo meniu elementai

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Atšaukti	Ctrl+Z	*Skaitmeninim*	*Undo and Redo*
Pakartoti	Ctrl+Shift+Z	*Skaitmeninim*	*Undo and Redo*
Iškirpti objektus	Ctrl+X	*Skaitmeninim*	*Cutting, Copying and Pasting Features*
Kopijuoti objektus	Ctrl+C	*Skaitmeninim*	*Cutting, Copying and Pasting Features*
Įkelti objektus	Ctrl+V	*Skaitmeninim*	*Cutting, Copying and Pasting Features*
Įkelti geoobjektus kaip ►			*Darbas su atributų lentele*
► Naują vektorinį sluoksnį…			*Darbas su atributų lentele*

continues on next page

Table 3.2 – tęsinys iš praeito puslapio

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
► *Laikiną juodraštinį sluoksnį..*	Ctrl+Alt+V		*Darbas su atributų lentele*
Trinti pažymėtus		*Skaitmeninim*	*Deleting Selected Features*
Pažymėti ►			*Geoobjektų pažymėjimas*
► *Pažymėti geoobjektą(us)*		*Pažymėjimo*	*Geoobjektų pažymėjimas*
► *Pažymėti geoobjektus poligonu*		*Pažymėjimo*	*Geoobjektų pažymėjimas*
► *Pažymėti geoobjektus laisvai*		*Pažymėjimo*	*Geoobjektų pažymėjimas*
► *Pažymėti geoobjektus spinduliu*		*Pažymėjimo*	*Geoobjektų pažymėjimas*
► *Pažymėti geoobjektus pagal reikšmę...*	F3	*Pažymėjimo*	*Geoobjektų pažymėjimas*
► *Pažymėti geoobjektus pagal išraišką...*	Ctrl+F3	*Pažymėjimo*	*Geoobjektų pažymėjimas*
► *Nužymėti geoobjektus visuose sluoksniuose*	Ctrl+Alt+A	*Pažymėjimo*	*Geoobjektų pažymėjimas*
► *Nužymėti geoobjektus dabartiniame sluoksnyje*	Ctrl+Shift+A	*Pažymėjimo*	*Geoobjektų pažymėjimas*
► *Iš naujo pažymėti geoobjektus*			*Geoobjektų pažymėjimas*
► *Pažymėti visus geoobjektus*	Ctrl+A	*Pažymėjimo*	*Geoobjektų pažymėjimas*
► *Invertuoti geoobjektų pažymėjimą*		*Pažymėjimo*	*Geoobjektų pažymėjimas*
Pridėti objektą	Ctrl+.	*Skaitmeninim*	
Pridėti taško geoobjektą	Ctrl+.	*Skaitmeninim*	*Adding Features*
Pridėti linijos geoobjektą	Ctrl+.	*Skaitmeninim*	*Adding Features*
Pridėti poligono geoobjektą	Ctrl+.	*Skaitmeninim*	*Adding Features*
Pridėti anotaciją ►			*Anotacijų įrankiai*
► *Formos anotacija*		*Anotacijos*	*Anotacijų įrankiai*
► *HTML anotacija*		*Anotacijos*	*Anotacijų įrankiai*
Keisti atributus ►			
► *Keisti pažymėtų geoobjektų atributus*		*Skaitmeninim*	*Editing attribute values*
► *Sulieti pažymėtų geoobjektų atributus*		*Sudėtingesnio skaitmenizavi*	*Merge attributes of selected features*
Keisti geometriją ►			
► *Perkelti geoobjektą(us)*		*Sudėtingesnio skaitmenizavi*	*Move Feature(s)*
► *Kopijuoti ir perkelti geoobjektą(us)*		*Sudėtingesnio skaitmenizavi*	*Move Feature(s)*
► *Pasukti geoobjektą(us)*		*Sudėtingesnio skaitmenizavi*	*Rotate Feature(s)*
► *Keisti geoobjekto(ų) dydį*		*Sudėtingesnio skaitmenizavi*	*Scale Feature*
► *Supaprastinti objektą*		*Sudėtingesnio skaitmenizavi*	*Simplify Feature*

continues on next page

Table 3.2 – tęsinys iš praeito puslapio

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
► *Pridėti žiedą*		*Sudėtingesnio skaitmenizavi*	*Add Ring*
► *Pridėti dalį*		*Sudėtingesnio skaitmenizavi*	*Add Part*
► *Užpildyti žiedą*		*Sudėtingesnio skaitmenizavi*	*Fill Ring*
► *Naikinti žiedą*		*Sudėtingesnio skaitmenizavi*	*Delete Ring*
► *Naikinti dalį*		*Sudėtingesnio skaitmenizavi*	*Delete Part*
► *Keisti geoobjektų formą*		*Sudėtingesnio skaitmenizavi*	*Reshape Features*
► *Kreivės postūmis*		*Sudėtingesnio skaitmenizavi*	*Offset Curves*
► *Padalinti objektus*		*Sudėtingesnio skaitmenizavi*	*Split Features*
► *Padalinti dalis*		*Sudėtingesnio skaitmenizavi*	*Split parts*
► *Sulieti parinktus geoobjektus*		*Sudėtingesnio skaitmenizavi*	*Merge selected features*
► *Viršūnės įrankis (visi sluoksniai)*		*Skaitmeninim*	*Vertex tool*
► *Viršūnės įrankis (dabartinis sluoksnis)*		*Skaitmeninim*	*Vertex tool*
► *Apsukti liniją*		*Sudėtingesnio skaitmenizavi*	*Reverse Line*
► *Patrumpinti/ištempti geoobjektą*		*Sudėtingesnio skaitmenizavi*	*Trim/Extend Feature*
Pasukti taškinius simbolius		*Sudėtingesnio skaitmenizavi*	*Rotate Point Symbols*
Atitraukti taško simbolius		*Sudėtingesnio skaitmenizavi*	*Offset Point Symbols*

Įrankiai, kurie priklauso nuo pažymėto sluoksnio geometrijos tipo, t.y. taškai, linijos ar poligonai, aktyvuojami pagal situaciją:

3.3 Table : „Perkelti geoobjektą“ nuo geometrijos priklausančios piktogramos

Meniu parinktis	Taškai	Linijos	Poligonai
Perkelti objektą(us)			
Kopijuoti ir perkelti geoobjektą(us)			

3.1.3 Vaizdas

Žemėlapis braižomas žemėlapio vaizduose. Jūs galite veikti šiuos vaizdus naudodami meniu *Rodymas* įrankius. Pavyzdžiui, jūs galite:

- Sukurti naujus 2D ar 3D žemėlapio vaizdus greta pagrindinės žemėlapio drobės
- *Keisti mastelį ar perstumti* vaizdą į bet kurią vietą
- Daryti rodomų geoobjektų atributų ar geometrijos užklausas
- Gerinti žemėlapio vaizdą peržiūros režimais, anotacijomis ar dekoracijomis
- Naudoti bet kurį skydelį ar įrankinę

Meniu taipogi jums leidžia keisti QGIS sąsają, naudojant veiksmus, tokius kaip:

- *Perjungti pilno ekrano režimą*: užima visą ekraną paslepiant antraštės juostą
- *Perjungti skydelių matomumą*: rodo ar slepia *skydelius* - naudinga skaitmeninant geoobjektus (geresniam drobės matomumui) bei (per projektorių rodomiems/įrašytiems) pristatymams naudojant QGIS pagrindinę drobę
- *Toggle Map Only*: hides panels, toolbars, menus and status bar and only shows the map canvas. Combined with the full screen option, it makes your screen display only the map.

3.4 Table : Vaizdo meniu elementai

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Naujas Žemėlapio vaizdas	`Ctrl+M`	*Žemėlapio navigacijos*	*2D žemėlapio vaizdas*
3D žemėlapio vaizdai ►			*3D žemėlapio vaizdas*
► *Naujas 3D žemėlapio vaizdas*	`Ctrl+Alt+M`	*Žemėlapio navigacijos*	*3D žemėlapio vaizdas*
► *Valdyti 3D žemėlapio vaizdus*			*3D žemėlapio vaizdas*
Paslinkti žemėlapį		*Žemėlapio navigacijos*	*Žemėlapio vaizdo tyrinėjimas*
Pastumti žemėlapį ties žymėjimu		*Žemėlapio navigacijos*	*Žemėlapio vaizdo tyrinėjimas*
Didinti	`Ctrl+Alt++`	*Žemėlapio navigacijos*	*Žemėlapio vaizdo tyrinėjimas*
Mažinti	`Ctrl+Alt+-`	*Žemėlapio navigacijos*	*Žemėlapio vaizdo tyrinėjimas*
Duomenų filtravimas ►			*2D žemėlapio vaizdas*
► *Laiko valdymas*		*Žemėlapio navigacijos*	*Laiko valdymo skydelis*
► *Aukščio valdymas*			*Aukščio valdymo skydelis*
Identifikuoti objektus	`Ctrl+Shift+I`	*Anotacijų*	*Geoobjektų identifikavimas*
Matuoti ►		*Anotacijų*	*Matavimas*
► *Matuoti liniją*	`Ctrl+Shift+M`	*Anotacijų*	*Matavimas*
► *Matuoti plotą*	`Ctrl+Shift+J`	*Anotacijų*	*Matavimas*
► *Matuoti kryptį*		*Anotacijų*	*Matavimas*
► *Matuoti kampą*		*Anotacijų*	*Matavimas*

continues on next page

Table 3.4 – tęsinys iš praeito puslapio

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Statistinė suvestinė		*Anotacijų*	*Statistinės suvestinės skydelis*
Aukščio profilis			*Aukščio profilio vaizdas*
Pilnai padidinti	Ctrl+Shift+F	*Žemėlapio navigacijos*	*Žemėlapio vaizdo tyrinėjimas*
Didinti iki pažymėjimo	Ctrl+J	*Žemėlapio navigacijos*	*Žemėlapio vaizdo tyrinėjimas*
Didinti iki sluoksnio(ių)		*Žemėlapio navigacijos*	*Žemėlapio vaizdo tyrinėjimas*
Didinti iki savos rezoliucijos (100%)		*Žemėlapio navigacijos*	*Žemėlapio vaizdo tyrinėjimas*
Didinti praėjusį		*Žemėlapio navigacijos*	*Žemėlapio vaizdo tyrinėjimas*
Didinti kitą		*Žemėlapio navigacijos*	*Žemėlapio vaizdo tyrinėjimas*
Dekoracijos ►	Alt+V + D		*Žemėlapio dekoravimas*
► *Tinklelis…*			*Tinklelis*
► *Mastelio juosta…*			*Mastelio juosta*
► *Paveikslas…*			*Piešinio dekoracija*
► *Šiaurės rodyklė…*			*Šiaurės rodyklė*
► *Antraštės etiketė…*			*Antraštės užrašas*
► *Autorių teisių etiketė…*			*Autorių teisių užrašas*
► *Sluoksnio apimtys…*			*Išdėstymų apimtys*
Peržiūros režimas ►			
► *Normalus*			
► *Imituoti monochrominį*			
► *Imituoti Achromatopsia spalvų nematymą (pilka)*			
► *Imituoti Protanopia spalvų neskyrimą (nėra raudonos)*			
► *Imituoti Deuteranopia spalvų neskyrimą (nėra žalios)*			
► *Imituoti Tritanopia spalvų neskyrimą (nėra mėlynos)*			
Rodyti žemėlapio patarimus		*Anotacijų*	*Display Properties*
Nauja erdvinė žymelė…	Ctrl+B	*Žemėlapio navigacijos*	*Žemėlapio žymelės apimčių*
Rodyti erdvines žymeles	Ctrl+Shift+B	*Žemėlapio navigacijos*	*Žemėlapio žymelės apimčių*
Rodyti erdvinių žymelių tvarkyklę			*Žemėlapio žymelės apimčių*
Atnaujinti	F5	*Žemėlapio navigacijos*	

continues on next page

Table 3.4 – tęsinys iš praeito puslapio

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Sluoksnio matomumas ►			*Sluoksnių skydelis*
► *Rodyti visus sluoksnius*	`Ctrl+Shift+U`		*Sluoksnių skydelis*
► *Slėpti visus sluoksnius*	`Ctrl+Shift+H`		*Sluoksnių skydelis*
► *Rodyti pažymėtus sluoksnius*			*Sluoksnių skydelis*
► *Slėpti pažymėtus sluoksnius*			*Sluoksnių skydelis*
► *Perjungti pažymėtus sluoksnius*			*Sluoksnių skydelis*
► *Perjungti pažymėtus sluoksnius nepriklausomai*			*Sluoksnių skydelis*
► *Slėpti nepažymėtus sluoksnius*			*Sluoksnių skydelis*
Skydeliai ►			*Skydeliai ir įrankinės*
Įrankinės ►			*Skydeliai ir įrankinės*
Perjungti pilno ekrano režimą	`F11`		
Perjungti skydelio matomumą	`Ctrl+Tab`		
Perjungti tik žemėlapį	`Ctrl+Shift+Tab`		

Naudojant Linux KDE, *Skydeliai* ►, *Įrankinės* ► ir *Perjungti pilno ekrano režimą* rasite *Nustatymų* meniu.

3.1.4 Sluoksnis

Sluoksnio meniu teikia daug įrankių, leidžiančių *kurti* naujus duomenų šaltinis, *pridėti* juos į projektą ar *įrašyti pakeitimus*. Naudodami tuos pačius duomenų šaltinius, jūs galite:

- *Dubliuoti* sluoksnį, kad sukurtumėte jo kopiją, kuriai galėsite pakeisti pavadinimą, stilių (simbologiją, etiketes, …), ryšius, … Kopija naudoja tą patį duomenų šaltinį, kaip ir originalas.
- *Kopijuoti* ir *Įkelti* sluoksnius ar grupes iš vieno projekto į kitą kaip naują egzempliorių, kurio savybes galite keisti nepriklausomai. Kaip ir *Dubliavimo* atveju, sluoksniai remsis tuo pačiu duomenų šaltiniu.
- arba *Įtraukti sluoksnius ir grupes…* iš kito projekto kaip tik skaitymo kopijas, kurių jūs negalite keisti (daugiau skaitykite *Išorinių projektų sluoksnių įtraukimas*)

Sluoksnių meniu taipogi leidžia konfigūruoti, kopijuoti ar įkelti sluoksnio savybes (stilių, mastelį, CRS…).

3.5 Table : Sluoksnio meniu elementai

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Duomenų šaltinių tvarkyklė	`Ctrl+L`	*Duomenų šaltinių tvarkyklės*	*Duomenų atvėrimas*
Sukurti sluoksnį ►			*Naujų vektorinių sluoksnių kūrimas*
► *Naujas GeoPackage sluoksnis…*	`Ctrl+Shift+N`	*Duomenų šaltinių tvarkyklės*	*Naujo GeoPackage sluoksnio kūrimas*
► *Naujas Shapefile sluoksnis…*		*Duomenų šaltinių tvarkyklės*	*Naujo Shape failų sluoksnio kūrimas*
► *Naujas SpatiaLite sluoksnis…*		*Duomenų šaltinių tvarkyklės*	*Naujo SpatiaLite sluoksnio kūrimas*
► *Naujas laikinas juodraštinis sluoksnis…*		*Duomenų šaltinių tvarkyklės*	*Creating a new Temporary Scratch Layer*

continues on next page

Table 3.5 – tęsinys iš praeito puslapio

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
► *Naujas tinklelio sluoksnis…*		*Duomenų šaltinių tvarkyklės*	*Creating a new Mesh layer*
► *Naujas GPX sluoksnis…*		*Duomenų šaltinių tvarkyklės*	*Creating a new GPX layer*
► *Naujas virtualus sluoksnis…*		*Duomenų šaltinių tvarkyklės*	*Creating virtual layers*
Pridėti sluoksnį ►			*Duomenų atidarymas*
► *Pridėti vektorinį sluoksnį……*	`Ctrl+Shift+V`	*Sluoksnių valdymo*	*Sluoksnio įkėlimas iš failo*
► *Pridėti rastro sluoksnį…*	`Ctrl+Shift+R`	*Sluoksnių valdymo*	*Sluoksnio įkėlimas iš failo*
► *Pridėti tinklelio sluoksnį…*		*Sluoksnių valdymo*	*Tinklelio sluoksnio įkėlimas*
► *Pridėti atskirtą teksto sluoksnį…*	`Ctrl+Shift+T`	*Sluoksnių valdymo*	*Atskirto tekstinio failo importavimas*
► *Add PostgreSQL Layer…*	`Ctrl+Shift+D`	*Sluoksnių valdymo*	*Su duomenų baze susiję įrankiai*
► *Pridėti SpatiaLite sluoksnį…*	`Ctrl+Shift+L`	*Sluoksnių valdymo*	*SpatiaLite sluoksniai*
► *Pridėti MS SQL Serverio sluoksnį…*		*Sluoksnių valdymo*	*Su duomenų baze susiję įrankiai*
► *Pridėti Oracle Spatial sluoksnį…*		*Sluoksnių valdymo*	*Su duomenų baze susiję įrankiai*
► *Pridėti SAP HANA Spatial sluoksnį…*		*Sluoksnių valdymo*	*Su duomenų baze susiję įrankiai*
► *Pridėti/keisti virtualų sluoksnį…*		*Sluoksnių valdymo*	*Creating virtual layers*
► *Pridėti WMS/WMTS sluoksnį…*	`Ctrl+Shift+W`	*Sluoksnių valdymo*	*Loading WMS/WMTS Layers*
► *Pridėti XYZ sluoksnį…*			*XYZ kaladėlių paslaugų naudojimas*
► *Pridėti WCS sluoksnį…*		*Sluoksnių valdymo*	*WCS Client*
► *Pridėti WFS / OGC API - Geoobjektų sluoksnį…*		*Sluoksnių valdymo*	*WFS and WFS-T Client*
► *Pridėti ArcGIS REST Server sluoksnį…*		*Sluoksnių valdymo*	*ArcGIS REST serverių naudojimas*
► *Pridėti vektorinių kaladėlių sluoksnį…*			*Vektorinių kaladėlių paslaugų naudojimas*
► *Pridėti taškų debesies sluoksnį…*			*Darbas su taškų debesimis*
► *Pridėti GPX sluoksnį…*			*Introducing GNSS/GPS Data*
Įtraukti sluoksnius ir grupes…			*Išorinių projektų sluoksnių įtraukimas*

continues on next page

Table 3.5 – tęsinys iš praeito puslapio

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Įdėti iš sluoksnio apibrėžimo failo…			*Sluoksnio apibrėžimo failas*
Geo-koordinatorius…			*Geo-koordinavimas*
Kopijuoti stilių			*Sluoksnio savybių įrašymas ir bendrinimas*
Įkelti stilių			*Sluoksnio savybių įrašymas ir bendrinimas*
Kopijuoti sluoksnį			
Įkelti sluoksnį/grupę			
Atverti atributų lentelę	`F6`	*Anotacijų*	*Darbas su atributų lentele*
Filtruoti atributų lentelę ►			*Darbas su atributų lentele*
► Atverti atributų lentelę (pažymėtus geoobjektus)	`Shift+F6`	*Anotacijų*	*Darbas su atributų lentele*
► Atverti atributų lentelę (matomus geoobjektus)	`Ctrl+F6`	*Anotacijų*	*Darbas su atributų lentele*
► Atverti atributų lentelę (pakeistus ir naujus geoobjektus)		*Anotacijų*	*Darbas su atributų lentele*
Perjungti redagavimą		*Skaitmeninimo*	*Digitizing an existing layer*
Įrašyti sluoksnio pakeitimus		*Skaitmeninimo*	*Saving Edited Layers*
Dabartiniai keitimai ►		*Skaitmeninimo*	*Saving Edited Layers*
► Įrašyti pažymėtam sluoksniui(iams)		*Skaitmeninimo*	*Saving Edited Layers*
► Atšaukti pažymėtam sluoksniui(iams)		*Skaitmeninimo*	*Saving Edited Layers*
► Nutraukti pažymėtam sluoksniui(iams)		*Skaitmeninimo*	*Saving Edited Layers*
► Įrašyti visiems sluoksniams		*Skaitmeninimo*	*Saving Edited Layers*
► Atšaukti visiems sluoksniams		*Skaitmeninimo*	*Saving Edited Layers*
► Nutraukti visiems sluoksniams		*Skaitmeninimo*	*Saving Edited Layers*
Įrašyti kaip…			*Creating new layers from an existing layer*
Įrašyti kaip sluoksnio apibrėžimo failą…			*Sluoksnio apibrėžimo failas*
Pašalinti sluoksnį/grupę	`Ctrl+D`		
Dubliuoti sluoksnį(-ius)			
Nustatyti sluoksnio(-ių) mastelio matomumą			*Nuo mastelio priklausomo matomumo parinkimas*
Nustatyti sluoksnio(ių) CRS	`Ctrl+Shift+C`		*Sluoksnio koordinačių atskaitos sistemos*
Nustatyti projekto CRS pagal sluoksnį			*Projekto koordinačių ataskaitos sistemos*
Sluoksnio savybės…			• *vector* • *raster* • *tinklelis* • *taškų masyvas* • *vektorinės kaladėlės*
Filtras…	`Ctrl+F`		*Query Builder*
Etiketės			*Labels Properties*

continues on next page

Table 3.5 – tęsinys iš praeito puslapio

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Rodyti peržiūroje			*Apžvalgos skydelis*
Viską rodyti peržiūroje			*Apžvalgos skydelis*
Slėpti viską peržiūroje			*Apžvalgos skydelis*

3.1.5 Nustatymai

3.6 Table : Nustatymų meniu elementai

Meniu parinktis	Aprašymas
Naudotojo profiliai ►	*Darbas su naudotojo profiliais*
► *default*	*Darbas su naudotojo profiliais*
► *Atverti aktyvaus profilio aplanką*	*Darbas su naudotojo profiliais*
► *Naujas profilis…*	*Darbas su naudotojo profiliais*
Stilių tvarkyklė…	*Stilių tvarkyklė*
Savos projekcijos…	*Sava koordinačių atskaitos sistema*
Klavišų kombinacijos…	*Klavišų kombinacijos*
Sąsajos tinkinimas…	*Tinkinimas*
Parinktys…	*Parinktys*

Jei naudojate Linux KDE, *Nustatymų* meniu rasite daugiau parinkčių, pavyzdžiui *Skydeliai* ►, *Įrankinės* ► ir *Įjungti pilno ekrano režimą*.

3.1.6 Priedai

3.7 Table : Priedų meniu elementai

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Tvarkyti ir diegti priedus…			*The Plugins Dialog*
Python'o konsolė	`Ctrl+Alt+P`	*Priedų*	*QGIS Python konsolė*

Pirmą kartą paleidus QGIS, įkeliami ne visi baziniai priedai.

3.1.7 Vektorius

The *Vector* menu provides tools for analyzing, managing and manipulating vector datasets. By default, it is filled with a set of predefined *Processing* algorithms, grouped by sub-menus. This provides shortcuts for many common vector-based GIS tasks from core providers. If not all these sub-menus are available, enable the *Processing* plugin in *Plugins* ► *Manage and Install Plugins…*.

Pastaba

The list of tools and their menu can be modified/extended with any Processing algorithms (read *Apdorojimo karkaso konfigūravimas*) or by some third-party *plugins*.

This is what the *Vector* menu looks like if all the core plugins are enabled.

3.8 Table : Vektoriaus meniu elementai

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Tikrinti geometrijas…			*Geometry Checker Plugin*
Topologijos tikrinimas		*Vektorių*	*Topology Checker Plugin*
Geoapdorojimo įrankiai ►	`Alt+O` + `G`		
► *Buferis…*			*Buferis*
► *Apkarpyti…*			*Apkirpti*
► *Išgaubtas korpusas…*			*Convex hull*
► *Skirtumas…*			*Skirtumas*
► *Sulieti…*			*Dissolve*
► *Susikirtimas…*			*Intersection*
► *Simetrinis skirtumas…*			*Symmetrical difference*
► *Sąjunga…*			*Union*
► *Naikinti pažymėtus poligonus…*			*Eliminate selected polygons*
Geometrijos įrankiai ►	`Alt+O` + `E`		
► *Centroidai…*			*Centroidai*
► *Rinkti geometrijas…*			*Collect geometries*
► *Ištraukti viršūnes…*			*Extract vertices*
► *Kelias dalis į vieną dalį…*			*Multipart to singleparts*
► *Linijas į poligonus…*			*Polygons to lines*
► *Paprastinti…*			*Simplify*
► *Tikrinti teisingumą…*			*Tikrinti tvarkingumą*
► *Delaunay trianguliacija…*			*Delaunay triangulation*
► *Tankinti skaičiumi…*			*Densify by count*
► *Pridėti geometrijos atributus…*			*Pridėti geometrijos atributus*
► *Linijas į poligonus…*			*Lines to polygons*
► *Voronoi poligonai…*			*Voronoi polygons*
Analizės įrankiai ►	`Alt+O` + `A`		
► *Linijų susikirtimai…*			*Line intersections*
► *Vidutinė(s) koordinatė(s)…*			*Mean coordinate(s)*
► *Pagrindinė laukų statistika…*			*Bazinė laukų statistika*
► *Skaičiuoti taškus poligone…*			*Skaičiuoti taškus poligone*
► *Atstumų matrica…*			*Distance matrix*
► *Rodyti unikalias reikšmes…*			*List unique values*
► *Artimiausio kaimyno analizė…*			*Nearest neighbour analysis*
► *Linijų ilgių suma…*			*Sum line lengths*
Duomenų valdymo įrankiai ►	`Alt+O` + `D`		

continues on next page

Table 3.8 – tęsinys iš praeito puslapio

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
► *Sulieti vektorinius sluoksnius…*			*Merge vector layers*
► *Perprojektuoti rastro sluoksnį…*			*Reproject layer*
► *Sukurti erdvinį indeksą…*			*Create spatial index*
► *Jungti atributus pagal poziciją…*			*Join attributes by location*
► *Dalinti vektorinį sluoksnį…*			*Split vector layer*
Tyrimo įrankiai ►	Alt+O + R		
► *Sukurti tinklelį…*			*Kurti tinklelį*
► *Ištraukti sluoksnio ribas…*			*Extract layer extent*
► *Atsitiktiniai taškai apimtyje…*			*Random points in extent*
► *Atsitiktiniai taškai poligonuose…*			*Random points in polygons*
► *Atsitiktiniai taškai ant linijų…*			*Random points on lines*
► *Pažymėti pagal vietą…*			*Select by location*
► *Parinkti pagal nuotolį…*			*Select within distance*
► *Atsitiktiniai taškai sluoksnio ribose…*			*Random points in layer bounds*
► *Atsitiktiniai taškai poligonuose…*			*Random points inside polygons*
► *Atsitiktinė parinktis…*			*Random selection*
► *Atsitiktinė parinktis poaibiuose…*			*Random selection within subsets*
► *Dėsningi taškai…*			*Regular points*

3.1.8 Rastras

The *Raster* menu provides tools for analyzing, managing and manipulating raster datasets. By default, it is filled with a set of predefined *Processing* algorithms, grouped by sub-menus. This provides shortcuts for many common raster-based GIS tasks from core providers. If not all these sub-menus are available, enable the *Processing* plugin in *Plugins* ► *Manage and Install Plugins…*.

Pastaba

The list of tools and their menu can be modified/extended with any Processing algorithms (read *Apdorojimo karkaso konfigūravimas*) or by some third-party *plugins*.

This is what the *Raster* menu looks like if all the core plugins are enabled.

3.9 Table : Rastro meniu elementai

Meniu parinktis	Kombinacija	Įrankin	Aprašymas
Rastro skaičiuotuvas…			*Raster Calculator*
Lygiuoti rastrus…			*Lygiuoti rastrus*
Analizė ►			
► *Aspektas…*			*Aspektas*
► *Užpildyti „nėraduomenų“…*			*Pildyti NėraDuomenų*

continues on next page

Table 3.9 – tęsinys iš praeito puslapio

Meniu parinktis	Kombinacija	Įrankin	Aprašymas
► *Tinklelis (judantis vidurkis)…*			*Grid (Moving average)*
► *Tinklelis (duomenų metrika)…*			*Grid (Data metrics)*
► *Tinklelis (atvirkštinis atstumas laipsniu)…*			*Grid (Inverse distance to a power)*
► *Tinklelis (artimiausias kaimynas)…*			*Grid (IDW with nearest neighbor searching)*
► *Beveik juoda…*			*Near black*
► *Šešėliavimas…*			*Hillshade*
► *Artumas (rastro atstumas)…*			*Proximity (raster distance)*
► *Šiurkštumas…*			*Roughness*
► *Sijoti…*			*Sieve*
► *Nuolydis…*			*Slope*
► *Topografinės pozicijos indeksas (TPI)…*			*Topographic Position Index (TPI)*
► *Paviršiaus šiurkštumo indeksas (TRI)…*			*Terrain Ruggedness Index (TRI)*
Projekcijos ►			
► *Priskirti projekciją…*			*Priskirti projekciją*
► *Ištraukti projekciją…*			*Ištraukti projekciją*
► *Sukraipyti (perprojektuoti)…*			*Kraipymas (perprojektavimas)*
Įvairūs ►			
► *Sukurti virtualų rastrą…*			*Kurti virtualų rastrą*
► *Rastro informacija…*			*Raster information*
► *Sulieti…*			*Sulieti*
► *Sukurti peržiūras (piramides)…*			*Kurti apžvalgas (piramides)*
► *Kaladėlių indeksas…*			*Tile index*
Ištraukimas ►			
► *Apkarpyti rastrą pagal apimtį…*			*Apkarpyti rastrą pagal apimtį*
► *Apkarpyti rastrą naudojant maskuotę…*			*Apkarpyti rastrą pagal šablono sluoksnį*
► *Kontūras…*			*Contour*
Keitimas ►			
► *PCT į RGB…*			*PCT į RGB*
► *Poligonizuoti (rastrą į vektorių)…*			*Polygonize (raster to vector)*
► *Rasterizuoti (vektorius į rastrą)…*			*Rasterize (vector to raster)*
► *RGB į PCT…*			*RGB to PCT*
► *Versti (keisti formatą)…*			*Translate (convert format)*

3.1.9 Duomenų bazės

The *Database* menu provides tools for Database interaction. If no database-related plugins are enabled, there will be no *Database* menu. This is what the *Database* menu looks like if all the core plugins are enabled.

3.10 Table : Duomenų bazės meniu elementai

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Keitimas neprisijungus…	Alt+D + O		*Offline Editing Plugin*
► *Keisti į neprisijungusį projektą…*		*Database*	*Offline Editing Plugin*
► *Sinchronizuoti*		*Database*	*Offline Editing Plugin*
Query History…		*Database*	*Creating SQL Query Layers*
DB tvarkyklė…		*Database*	*DB Manager Plugin*

Pirmą kartą paleidus QGIS, įkeliami ne visi baziniai priedai.

3.1.10 Tinklas

Taip atrodo meniu *Tinklas*, kai įjungti visi baziniai meniu. Jei neįjungti jokie priedai, tai nebus ir meniu *Tinklas*.

3.11 Table : Tinklo meniu elementai

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
MetaSearch ►	Alt+W + M		*MetaSearch Catalog Client*
► *Metasearch*		*Žiniatinklio*	*MetaSearch Catalog Client*
► *Pagalba*			*MetaSearch Catalog Client*

Pirmą kartą paleidus QGIS, įkeliami ne visi baziniai priedai.

3.1.11 Tinklelis

Meniu *Tinklelis* teikia įrankius, kurių reikia ref:*tinklelio sluoksniams <label_meshdata>*. Trečiųjų šalių priedai gali pridėti šio meniu elementų.

3.12 Table : Tinklelio meniu elementai

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Tinklelio skaičiuotuvas…			*Mesh Calculator*
Perindeksuoti paviršius ir viršūnes			*Reindexing meshes*

3.1.12 Apdorojimas

3.13 Table : Apdorojimo meniu elementai

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Įrankinė	`Ctrl+Alt+T`		*Įrankinė*
Modelių dizaineris…	`Ctrl+Alt+G`		*Modelių dizaineris*
Istorija…	`Ctrl+Alt+H`		*The history manager*
Rezultatų žiūryklė	`Ctrl+Alt+R`		*Išorinių aplikacijų konfigūravimas*
Keisti geoobjektus vietoje			*The Processing in-place layer modifier*

Pirmą kartą paleidus QGIS, įkeliami ne visi baziniai priedai.

3.1.13 Pagalba

3.14 Table : Pagalbos meniu elementai

Meniu parinktis	Kombinacija	Įrankinė	Aprašymas
Pagalbos turinys	`F1`	*Pagalba*	
API dokumentacija			
Priedai ►			
Pranešti apie problemą			
Reikia komercinio palaikymo?			
QGIS namų puslapis	`Ctrl+H`		
Tikrinti QGIS versiją			
Apie			
QGIS palaikantys nariai			

3.1.14 QGIS

Šį meniu rasite tik macOS ir jame yra kai kurios su OS susijusios komandos.

3.15 Table : QGIS meniu elementai

Meniu parinktis	Kombinacija	Aprašymas
Apie QGIS		
Parinktys	`Cmd+,`	*Parinktys*
Slėpti QGIS	`Cmd+H`	
Rodyti viską		
Slėpti kitus	`Shift+Cmd+H`	
Išjungti QGIS	`Cmd+Q`	

Parinktys atitinka *Nustatymai* ► *Parinktys*, *Apie QGIS* atitinka *Pagalba* ► *Apie*, o *Išjungti QGIS* atitinka *Projektas* ► *Išjungti QGIS*.

3.2 Skydeliai ir įrankinės

Iš menu *Rodymas* (arba *Nustatymai*), jūs galite įjungti ir išjungti QGIS valdiklius (*Skydeliai* ►) ir įrankines (*Įrankinės* ►). Norėdami perjungti bet kurį iš jų, spauskite dešinį pelės mygtuką meniu juostoje ar įrankinėje ir parinkite jums reikiamą elementą. Skydeliai ir įrankinės gali būti perkeltos ir padėtos ten, kur jums reikia QGIS sąsajoje. Sąrašą taipogi galima plėsti įjungiant *Bazinius ar išorinius priedus*.

3.2.1 Įrankinės

Įrankinės suteikia prieigą prie daugumos meniu funkcijų bei prie papildomų įrankių sąveikai su žemėlapiu. Kiekvienas įrankinės elementas turi iššokančią pagalbą. Užveskite pelės žymeklį virš elemento ir bus parodytas trumpas įrankio tikslo aprašymas.

Galimos įrankinės:

3.16 Table : QGIS įrankinės

Įrankinės pavadinimas	Pagrindinė įrankių informacija
Sudėtingesnio skaitmenizavimo	*Advanced digitizing*
Anotacijos	*Anotacijų įrankiai*
Anotacijų	*Darbas su atributų lentele*, *Bendri įrankiai*
Duomenų šaltinių tvarkyklės	*Duomenų šaltinių valdymas*
Database	*DB Manager Plugin*
Skaitmeninimo	*Digitizing an existing layer*
GPS	*Live GPS tracking*
GRASS	*GRASS GIS Integration*
Pagalba	
Etiketė	*The Label Toolbar*
Sluoksnių valdymo	*Duomenų atidarymas*
Žemėlapio navigacijos	*Žemėlapio vaizdo tyrinėjimas*
Tinklelio skaitmenizavimas	*Editing a mesh layer*
Priedų	*Priedai*
Projekto	*Darbas su projekto failais*, *Laying out the maps*, *Stilių biblioteka*
Apdorojimo algoritmai	*Apdorojimo karkaso konfigūravimas*
Rastro	*Priedai*
Pažymėjimo	*Geoobjektų pažymėjimas*
Formų skaitmeninimas	*Shape digitizing*
Pritraukimas	*Setting the snapping tolerance and search radius*
Vektorių	*Priedai*
Žiniatinklio	*Priedai*, *MetaSearch Catalog Client*

Pastaba

Trečiųjų šalių priedai gali išplėsti pradinę įrankinę savo įrankiais arba teikti savo atskirą įrankinę.

Patarimas

Įrankinių atstatymas

Jeigu netyčia paslėpėte įrankinę, jūs galite ją grąžinti naudodami *Rodymas* ► *Įrankinės* ► (ar *Nustatymai* ► *Įrankinės* ►). Jei, dėl kokios nors priežasties, įrankinė (ar bet koks kitas valdiklis) visiškai dingsta iš naudotojo sąsajos, patarimų rasite *pradinio GUI atstatymas*.

3.2.2 Skydeliai

QGIS teikia daug skydelių. Skydeliai yra specialūs valdikliai, su kurias jūs galite sąveikauti (pažymėti parinktis, įjungti varneles, pildyti reikšmes…) vykdydami sudėtingas užduotis.

Žemiau yra numatytųjų QGIS skydelių sąrašas:

3.17 Table : QGIS skydeliai

Skydelio pavadinimas	Kombinacija	Aprašymas
Sudėtingesnio skaitmenizavimo	`Ctrl+4`	*The Advanced Digitizing panel*
Naršyklė	`Ctrl+2`	*Naršyklės skydelis*
Naršyklė (2)		*Naršyklės skydelis*
Tinkinimo/kūrimo įrankiai	`F12`	*Tinkinimo/kūrimo įrankių skydelis*
Aukščio profilis		*Aukščio profilio vaizdas*
Geometrijos tikrinimas		*Digitizing Properties*
GPS informacija	`Ctrl+0`	*Live GPS tracking*
GRASS įrankiai		*GRASS GIS Integration*
Sluoksnių eiliškumas	`Ctrl+9`	*Sluoksnių rikiuotės skydelis*
Sluoksnio stilius	`Ctrl+3`	*Sluoksnio stiliaus skydelis*
Sluoksniai	`Ctrl+1`	*Sluoksnių skydelis*
Pranešimų žurnalas		*Pranešimų žurnalo skydelis*
Peržiūra	`Ctrl+8`	*Apžvalgos skydelis*
Apdorojimo įrankinė		*Įrankinė*
Rezultatų žiūryklė		*Įrankinė*
Pritraukimo ir skaitmeninimo parinktys		*Setting the snapping tolerance and search radius*
Erdvinių žymelių tvarkyklė	`Ctrl+7`	*Žemėlapio apimčių žymelės*
Statistika	`Ctrl+6`	*Statistinės suvestinės skydelis*
Laiko valdymas		*Laiko valdymo skydelis*
Kaladėlės mastelis		*Kaladėlių rinkiniai*
Atšaukti/pakartoti	`Ctrl+5`	*Atšaukimo/pakartojimo skydelis*
Viršūnių redaktorius		*The Vertex Editor Panel*

3.3 Būsenos juosta

Būsenos juosta suteikia jums bendrą informaciją apie žemėlapio vaizdą ir apdorojamus ar galimus veiksmus, taipogi siūlo jums įrankius žemėlapio vaizdo tvarkymui.

3.3.1 Vietos juosta

Kairėje būsenos juostos pusėje yra vietos juosta - greitos paieškos valdiklis, padedantis jums rasti ir vykdyti bet kurią QGIS savybę ar parinktį:

1. Spauskite teksto valdiklį ar spauskite `Ctrl+K`, kad aktyvuotumėte vietos paieškos juostą.
2. Įrašykite tekstą, susijusį su jūsų ieškomu elementu (pavadinimą, žymą, raktažodį, …). Pagal nutylėjimą rezultatai bus grąžinami įjungtiems lokatoriaus filtrams, bet jūs galite riboti paiešką tik konkrečioje apimtyje jūsų teksto pradžioje parašę *lokatoriaus filtrų* priešdėlį, pvz. parašius `l cad` grąžins tik sluoksnius, kurių pavadinime yra `cad`.

   Filtrą taipogi galima pažymėti dvigubu paspaudimu meniu, kuris rodomas atidarant lokatoriaus valdiklį.
3. Spauskite rezultatą, kad įvykdytumėte atitinkamą veiksmą, priklausomai nuo elemento tipo.

Patarimas

Apribokite paiešką konkrečiam aktyvaus sluoksnio laukui(ams)

Pagal nutylėjimą paieška su filtru(`f`) „aktyvaus sluoksnio geoobjektai“ vykdoma visuose atributų lentelės laukuose. Jūs galite apriboti konkrečiu lauku naudodami priešdėlį `@`. Pvz., `f @pavadinimas par` arba `@pavadinimas par` grąžins tik geoobjektus, kurių atribute „pavadinimas“ yra „par“. Rašant įjungiamas automatinis teksto pabaigimas, pasiūlymą galima pritaikyti paspaudus mygtuką `Tab`.

Išmanesnis paieškoje naudojamų laukų valdymas galimas iš sluoksnio kortelės *Laukai*. Daugiau informacijos rasite *Fields Properties*.

Paieška valdoma naudojant gijas, taigi rezultatai matomi iš karto, kai tik galima, net jei įdiegti lėti paieškos filtrai. Rezultatų įrašai matosi iš karto, kai aptinka filtras, tai reiškia, kad pvz. failo paieškos filtras grąžins rezultatus po vieną, skenuojant failų medį. Tai užtikrina, kad naudotojo sąsaja visada reaguoja, net jei vykdoma labai lėta paieška (pvz. naudojanti internetinę paslaugą).

Pastaba

Nominatim paieškos įrankis gali veikti skirtingai (be paieškos automatinio užbaigimo, uždelsimas atsiunčiant rezultatus ir pan.) priklausomai nuo OpenStreetMap Nominatim naudojimo sąlygų.

Patarimas

Greita prieiga prie lokatoriaus konfigūracijos

Spauskite piktogramą lokatoriaus valdiklio viduje būsenos juostoje, kad parodytumėte sąrašą filtrų, kuriuos galite naudoti ir įrašą *Konfigūruoti*, kuris atidaro *Nustatymai* ► *Parinktys…* meniu kortelę *Lokatorius*.

3.3.2 Veiksmų pranešimai

Plote greta paieškos juostos yra jūsų įvykdytų veiksmų suvestinė, kuri rodoma tada, kai reikia (pvz. žymint geoobjektus sluoksnyje, išimant sluoksnį, žemėlapio pastūmimo atstumas ir kryptis) arba ilgas aprašymas įrankio, ant kurio jūs užvedėte pelės žymeklį (veikia ne visiems įrankiams).

Ilgų veiksmų, tokių kaip statistikos rinkimo rastro sluoksniuose, apdorojimo algoritmų vykdymo ar kelių sluoksnių braižymo žemėlapio vaizde, atvejais, bus rodoma eigos juosta.

3.3.3 Žemėlapio drobės valdymas

Koordinatės parinktis rodo dabartinę pelės žymeklio poziciją, seka jos judėjimą žemėlapyje. Jūs galite nurodyti vienetus (ir tikslumą) kortelėje *Projektas* ► *Savybės…* ► *Bendra*. Spauskite mažą mygtuką kairėje tekstinio lauko pusėje, kad perjungtumėte tarp Koordinačių parinkties ir *Apimties* parinkties, rodančios dabartinio žemėlapio vaizdo apatinio-kairio ir viršutinio-dešinio kampų koordinates žemėlapio vienetais.

Greta koordinačių rodymo rasite *Mastelį*. Jis rodo žemėlapio vaizdo mastelį. Jame yra ir mastelio parinkiklis, leidžiantis pasirinkti iš *iš anksto parinktų mastelių*.

Spauskite mygtuką dešinėje mastelio pusėje, kad užrakintumėte mastelį ir naudotumėte didinimo įrankį priartinimui ir nutolinimui. Didinimas leidžia priartinti nekeičiant žemėlapio mastelio, taip lengviau tiksliau tinkinti etikečių ir simbolių padėtį. Didinimo lygis išreiškiamas procentais. Jei *Padidinimo* lygis yra 100%, tai dabartinis žemėlapis nedidinamas, t.y. jis braižomas tiksliai pagal mastelį ir vaizduoklio rezoliuciją (DPI). Numatytoji didinimo reikšmė gali būti nurodyta *Nustatymai* ► *Parinktys* ► *Braižymas* ► *Braižymo elgsena*. Tai labai naudinga didelės

raiškos ekranams, kur galima padidinti mažus simbolius. Papildomai nustatymas *Nustatymai* ► *Parinktys* ► *Drobė ir legenda* ► *DPI* nurodo ar QGIS atsižvelgia į kiekvieno monitoriaus fizinį DPI ar naudoja bendrą sistemos loginį DPI.

Didinimo įrankio dešinėje galite nurodyti dabartinį žemėlapio vaizdo pasukimą laipsniais laikrodžio rodyklės kryptimi.

Būsenos juostos dešinėje, varnelę *Braižyti* galima naudoti norint laikinai sustabdyti žemėlapio vaizdo braižymą (daugiau rasite skyriuje *Žemėlapio braižymo kontrolė*).

Dešinėje nuo varnelės *Braižyti*, jūs rasite mygtuką *EPSG:kodas*, kuris rodo dabartinio projekto CRS. Paspaudus šį mygtuką atidaromas dialogas *Projekto savybės*, kuriame galite perprojektuoti žemėlapio vaizdą ar pakeisti bet kurią kitą projekto savybę.

Patarimas

Teisingo jūsų žemėlapio drobės mastelio skaičiavimas

Paleidus QGIS, numatytoji CRS yra `WGS 84 (EPSG 4326)`, o vienetai yra laipsniai. Tai reiškia, kad QGIS interpretuos bet kokias jūsų sluoksnio koordinates kaip laipsnius. Norėdami gauti teisingas mastelio reikšmes, jūs galite arba rankomis pakeisti šį nustatymą kortelėje *Bendra Projektas* ► *Savybės…* (pvz. į metrus), arba galite naudoti EPSG:code piktogramą, matytą aukščiau. Paskutiniu atveju vienetai nustatomi tokie, kaip nurodo projekto projekcija (pvz., `+units=us-ft`).

Pastebėtina, kad pradinį CRS galima nustatyti *Nustatymai* ► *Parinktys* ► *CRS valdymas*.

3.3.4 Pranešimai

Mygtukas Pranešimai atidaro *Pranešimų žurnalo skydelį*, kuriame yra informacija apie procesus (QGIS paleidimą, priedų įkėlimą, apdorojimo įrankius…)

SKYRIUS 4

Naršyklės skydelis

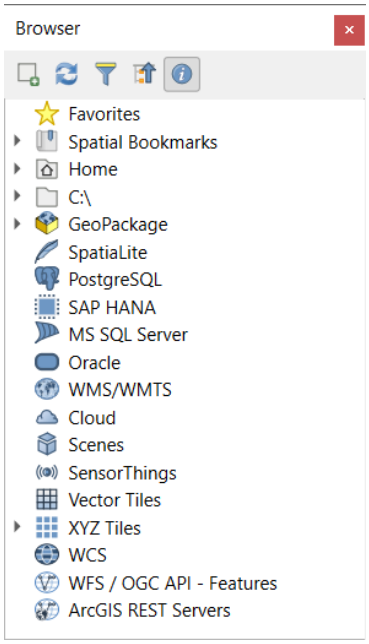
- *Resursai, kurie gali būti atidaryti/paleisti naršyklėje*
- *Naršyklės skydelio aukščiausio lygio įrašai*
  - *Parankiniai*
  - *Erdvinės žymelės*
  - *Projekto namų aplankas*
  - *Diskai ir failų sistema*
  - *Duomenų tipas*
  - *Duomenų bazių įrašai*
  - *Kaladėlės ir Žiniatinklio paslaugos*
  - *Projektas ir palaikomi failų tipai*

QGIS Naršyklės skydelis yra puikus įrankis QGIS resursų naršymui, paieškai, tikrinimui, kopijavimui bei įkėlimui. Naršyklėje rodomi tik tie resursai, su kuriais QGIS moka dirbti.

Naudodami Naršyklės skydelį jūs galite surasti, tikrinti ir pridėti duomenis, kaip aprašyta skyriuje *Naršyklės skydelis*. Papildomai naršyklės skydelis leidžia tempti ir numesti įvairius QGIS resursus, tokius kaip projekto failus, Python scenarijus ir apdorojimo modelius.

Python scenarijus, apdorojimo scenarijus ir apdorojimo modelius taipogi galima atidaryti redagavimui išoriniame redaktoriuje ir su grafiniu modeliuotoju.

You can drag and drop layers from the *Layers* panel to the *Browser* panel, for instance into a GeoPackage or a PostgreSQL database.



4.1 Fig. : Naršyklės skydelis

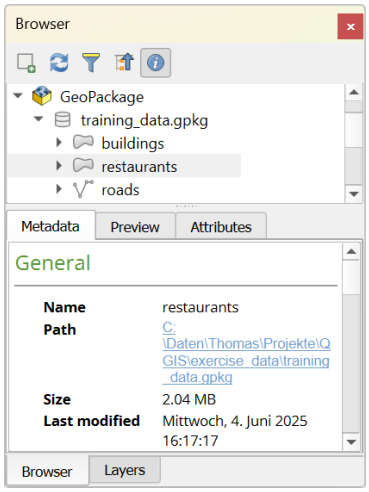
Naršyklės skydelis (Fig. 4.1) organizuojamas kaip išplečiama hierarchija su keliais fiksuotais viršutinio lygio elementais, kurie organizuoja naršyklės palaikomus resursus. Vaikiniai įrašai išplečiami paspaudus mygtuką , kurį rasite kairiau įrašo pavadinimo. Šaka suskleidžiama paspaudus . Mygtukas Sutraukti viską suskleidžia visus viršutinio lygio elementus.

Nustatymai ► *Sąsajos tinkinimas* galima išjungti resursus. Jei jūs, pavyzdžiui, nenorite naršyklėje matyti Python scenarijų, jūs galite išjungti įrašą *Browser* ► *py*, o jei norite atsikratyti savo namų aplanku naršyklėje, galite išjungti įrašą *Browser* ► *special:Home*.

Paieškai pagal pavadinimą (tiek hierarchijos šakų, tiek ir lapų paieškai) galite naudoti filtrą (Naršyklės filtras). Naudodami iškrentantį meniu Parinktys jūs galite:

- perjungti parinktį *Skirti raidžių dydį*
- parinkti vieną iš *Filtro sintaksių*
  - *Normali*
  - *Pakaitos simbolis(iai)*
  - *Reguliari išraiška*

Savybių valdiklis rodo naudingą informaciją apie kai kuriuos įrašus/resursus, gali būti įjungtas/išjungtas mygtuku Įjungti/išjungti savybių valdiklį. Įjungus atsidaro naršyklės skydelio apačioje, kaip parodyta Fig. 4.2.



4.2 Fig. : Savybių skydelis

Antrą naršyklės skydelį galima įjungti aktyvavus *Naršyklė (2)* skydelį *Rodymas* ► *Skydeliai*. Du naršyklės skydeliai gali praversti kopijuojant sluoksnius tarp resursų, kurie yra giliai skirtingų hierarchijos šakų viduje.

4.1 Resursai, kurie gali būti atidaryti/paleisti naršyklėje

Daug ką galima daryti naršyklėje:

- Pridėti vektorių, rastro ir tinklelio sluoksnius į jūsų žemėlapį du kartus paspaudus, pertempus į žemėlapio drobę ar paspaudus mygtuką Pridėti pažymėtus sluoksnius (prieš tai pažymėjus sluoksnius)
- Vykdyti Python scenarijus (įskaitant Apdorojimo algoritmus) du kartus paspaudus ar pertempus į žemėlapio drobę
- Vykdyti modelius du kartus paspaudus ar pertempiant į žemėlapio drobę
- *Ištraukti simbolius…* iš QGIS projekto failų naudojant kontekstinį meniu
- Atverti failus su jų numatyta programa (*Atverti <file type> išorėje…* kontekstiniame meniu). Pavyzdžiai: HTML failai, skaičiuoklės, piešiniai, PDF, tekstiniai failai, …
- Kopijuoti įrašus
- Pervadinti ir trinti (kelis) sluoksnius (kontekstinis meniu: *Tvarkyti* ►)
- Atverti failų tyrėjo langą ir tiesiogiai parinkti failą *Rodyti failuose*

Konkretiems resursams tinkantys veiksmai rodomi skirtingoms resursų grupėms išrikiuoti po žemiau išvardintais aukščiausio lygio įrašais.

4.2 Naršyklės skydelio aukščiausio lygio įrašai

4.2.1 Parankiniai

Dažnai naudojamos failų sistemos vietos gali būti pažymėtos kaip parankiniai. Čia bus rodomos jūsų pažymėtos vietos.

Be veiksmų, aprašytų „Namų aplanke“, šis kontekstinis meniu leidžia jums *Pervadinti parankinį…* ir *Pašalinti parankinį*.

4.2.2 Erdvinės žymelės

This is where you will find your *spatial bookmarks*, divided into *Project Bookmarks* and *User Bookmarks*.

4.1 Table : available actions for spatial bookmarks

Lygis	Contextual menu
Spatial Bookmarks	*New Spatial Bookmark…*
	Show the Spatial Bookmark Manager
	Import Spatial Bookmarks…
	Export Spatial Bookmarks…
Bookmark group (*Project Bookmarks* and *User Bookmarks*)	*New Spatial Bookmark…*
	Import Spatial Bookmarks…
	Export Spatial Bookmarks…
Individual bookmark	*Zoom to Bookmark*
	Edit Spatial Bookmark…
	Delete Spatial Bookmark

4.2.3 Projekto namų aplankas

Rodomas tada, kai projekto failas jau įrašytas. Įrašas *Projekto namų aplankas* yra aplankas, kuriame yra visi duomenys ir kitas turinys (skriptai, modeliai, tekstai, …), kuris gali būti naudojamas dabartiniame projekte. Rodomas *Naršyklės* skydelyje jis leidžia jums greitai pasiekti duomenis ir kitus projekto failus.

Pagal nutylėjimą tai projekto failo aplankas, bet jį galima pakeisti per parinktį *Projektas* ► *Savybės…* ► *Bendra* ► *Projekto namai* arba Naršyklės skydelyje paspaudus dešinį pelės mygtuką ant *Projekto namų aplanko* elemento ir parinkus *Nustatyti projekto namų aplanką…*. Tinkinti šį aplanką ypač naudinga, kai QGIS projektai nelaikomi šakniniame organizacijos „projektų“ aplanke kartu su duomenų rinkiniais.

4.2.4 Diskai ir failų sistema

Kiti *Naršyklės* skydelio elementai priklauso nuo naudojamos OS ir susiję su failų sistemos pagrindiniais elementais.

Jie pagrinde yra:

- *Namų* aplankas, rodantis į dabartinio naudotojo namų aplanką
- Unix-tipo mašinose tai šakninis `/` aplankas
- prijungti vietiniai arba tinklo diskai. Priklausomai nuo OS, jie rodomi tiesiogiai (pvz., `C:\`, `D:\`) arba per įrašą `/Volumes`.

Iš kiekvieno šių aplankų ir diskų kontekstinio meniu jūs galite:

- atnaujinti turinį
- sukurti *Naują* ► vidinį elementą, kuris yra *Aplankas*, *GeoPackage* arba ESRI *Shapefile* formato duomenų rinkinys
- paslėpti aplanką (*Slėpti iš naršyklės*)
- *Nustatyti spalvą*: keisti aplanko piktogramos spalvą, leidžiant greičiau naršyti per sudėtingą aplankų struktūrą
- įjungti *Skenavimą*:
  - *Stebėti pakeitimus*: leidžia valdyti, ar reikia sekti pasikeitimus konkrečiame aplanke. Šis nustatymas taikomas parinktam aplankui ir visiems poaplankiams. Tai reiškia, kad jūs galite rankiniu būdu nurodyti stebėti tinklo diskus, jei žinote, kad tai nėra problema, arba rankiniu būdu nurodyti nebestebėti didelių aplankų, kurių nenorite stebėti dėl kitų priežasčių. Pagal nutylėjimą nutolę ar tinklo diskai automatiškai nestebimi.

  - *Greitai nuskaityti šį aplanką*
- atverti aplanką jūsų failų tvarkyklėje (*Atverti aplanką…*)
- atverti aplanką terminalo lange (*Atverti terminale…*)
- patikrinti *Savybes…* ar tėvinio *Aplanko savybes…*

4.2.5 Duomenų tipas

4.2 Table : Available actions for plain layer files

Level and Context menu	Sluoksnių tipai Raster (*Reference*)	Vector (*Reference*)	Point clouds (*Reference*)	Mesh (*Reference*)
File				
Trinti failą <dataset name>	☑	☑		
Rodyti failuose	☑	☑	☑	☑
Tvarkyti	☑	☑	☑	☑
File Properties…	☑	☑	☑	☑
Open <file type> Externally…	☑			
Layer				
Pridėti sluoksnį į projektą	☑	☑	☑	☑
Eksportuoti sluoksnį	☑	☑		
Layer Properties…	☑	☑	☑	☑
Open with Data Source Manager…	☑	☑		
Execute SQL…		☑		

4.2.6 Duomenų bazių įrašai

Priklausomai nuo jūsų OS ir įdiegtų tvarkyklių, jūs galite turėti prieigą prie skirtingų duomenų bazių tipų naudojimui QGIS. Žemiau pateikiamas sąrašas skirtingų kontekstinio meniu įrašų kiekvienam duomenų aibių medžio lygiui.

4.3 Table : Available actions for database entries

Level and Context menu	Duombazės tipas GeoPackage (1)	SpatiaLite	PostgreSQL	SAP HANA	MS SQL Server	Oracle
Top menu						
Kurti *Naują jungtį…* į esamą duomenų bazę	☑	☑	☑	☑	☑	☑
Kurti duombazę…	☑	☑				
Kurti duombazę ir sluoksnį…	☑					
Įrašyti jungčių… detales į failą			☑	☑	☑	

continues on next page

Table 4.3 – tęsinys iš praeito puslapio

Įkelti jungtis…			✓	✓	✓	
Connection / Database						
Atnaujinti jungtį			✓	✓	✓	✓
Keisti jungties… nustatymus			✓	✓	✓	✓
Dubliuoti jungtį			✓	✓	✓	✓
Išimti jungtį…	✓	✓	✓	✓	✓	✓
Import Vector Layer…	✓	✓	✓	✓	✓	
Naujas lauko domenas ► Naujas intervalo domenas	✓					
Naujas lauko domenas ► Naujas reikšmių kodų domenas	✓					
Naujas lauko domenas ► Naujas Glob domenas	✓					
Trinti <database_name>	✓	✓				
Suspausti duombazę (VACUUM)	✓					
Kurti *Naują schemą…*			✓	✓	✓	
Kurti *Naują lentelę…*	✓	✓	✓	✓		
Vykdyti SQL… užklausą	✓	✓	✓	✓		
Schema						
Atnaujinti schemą			✓	✓	✓	
Schemos veiksmai ► Pervadinti schemą…			✓	✓	✓	
Schemos veiksmai ► Trinti schemą…			✓	✓	✓	
Kurti *Naują lentelę…*			✓	✓		
Vykdyti SQL… užklausą			✓	✓		
Table / Layer						
Lentelės veiksmai ► Pervadinti lentelę…			✓	✓	✓	
Lentelės veiksmai ► Ištuštinti lentelę…			✓		✓	
Vykdyti SQL… užklausą	✓	✓	✓			
Eksportuoti sluoksnį ► Į failą…	✓	✓	✓	✓	✓	
Tvarkyti ► Pervadinti sluoksnį <layer_name>…	✓	✓			✓	
Tvarkyti ► Trinti sluoksnį <layer_name>…	✓	✓	✓	✓	✓	
Tvarkyti ► Trinti pažymėtus sluoksnius	✓	✓	✓	✓	✓	
Tvarkyti ► Pridėti sluoksnį į projektą	✓	✓	✓	✓	✓	

continues on next page

Table 4.3 – tęsinys iš praeito puslapio

Tvarkyti ► Pridėti pažymėtus sluoksnius į projektą	✓	✓	✓	✓	✓
Manage ► Move to Another Schema			✓		✓
Atverti *Sluoksnio savybių…* dialogą	✓	✓	✓	✓	✓
Atverti *Failo savybių…* dialogą	✓				
Atverti su duomenų šaltinių tvarkykle	✓	✓			
Fields					
Pridėti naują lauką…	✓	✓	✓	✓	
Field					
Nustatyti lauko domeną	✓				
Rename Field ([2])	✓		✓		✓
Nustatyti sinonimą…	✓				
Nustatyti komentarą…	✓		✓		
Trinti lauką…	✓	✓	✓	✓	

Pastaba

Įkeliant vektorinius sluoksnius į QGIS, laukai su *Laukų domenais* (tokiais kaip apibrėžtais GeoPackage ar ESRI File Geodatabase) aptinkami automatiškai. Šie domenai yra duomenų bazės lygio apribojimai, tai reiškia juos užtikrina pati duomenų bazė ir tai taikoma visur, ne tik QGIS.

Importing vector layers into databases using Browser panel

QGIS allows you to import vector layers directly into a database from the *Browser* panel. This provides a simple way to transfer datasets into supported database formats without using DB Manager (see *DB Manager Plugin*).

You can start the import in two ways:

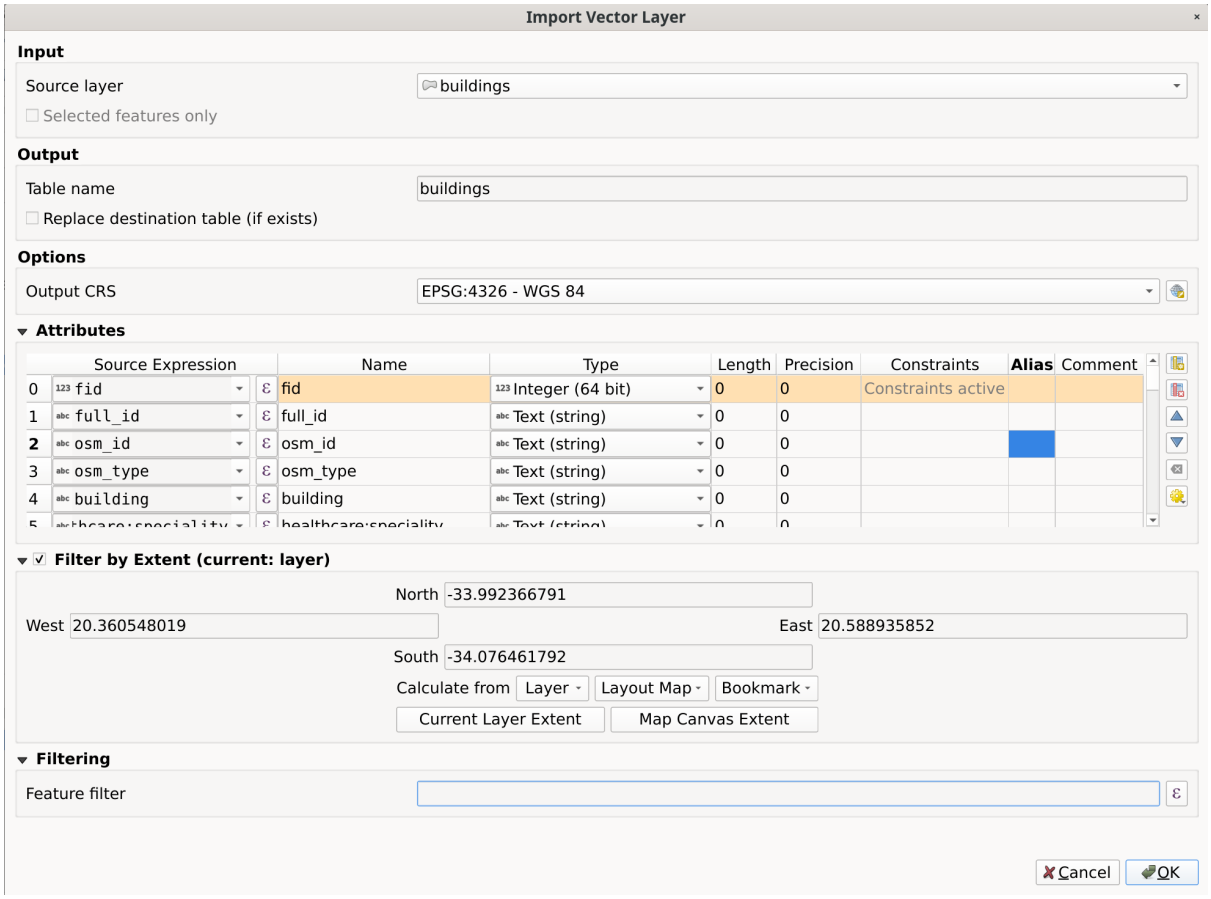
- Drag and drop a single vector layer from the Browser panel or from the Project into a database connection or schema item.
- Right-click on a database connection or schema item and choose *Import Vector Layer…* from the context menu.

Pastaba

When multiple layers are dragged into a database, they are imported immediately without further options. All imports run in the background, so the QGIS interface remains responsive during the process.

[1] Skirtingi įrašai gali pasirodyti kai kuriems GDAL palaikomiems vektorinių failų formatams, tokiems kaip ESRI File Geodatabase, FlatGeobuf, GeoParquet, NetCDF, … kai jie suderinami.

[2] For PostgreSQL connections: only supported with ALTER TABLE privilege.



4.3 Fig. : Import Vector Layer dialog

To add a layer using *Import Vector Layer…* follow these steps:

1. Add *Source layer*: the layer to import, chosen from the current QGIS project. To import only selected features of the source layer use the *Selected features only* option.
2. Set the *Table name*: the name of the destination table in the database. By default, it is taken from the source layer name but can be modified. You can also use the *Replace destination table (if exists)* to overwrite an existing table.
3. Define the *Output CRS*: the coordinate reference system of the destination table. By default, it is the CRS of the source layer, but it can be changed using the *CRS selector*.
4. Configure the *Attributes*:
   - *Source expression*: expression used to populate the field. By default, this corresponds to the original source field.
   - *Name*: name of the field in the destination table.
   - *Type*: field data type (integer, text, real, etc.).
   - *Length* and *Precision*: length and precision of the field, where applicable.
   - *Constraints*: rules to enforce on the field values (e.g. NOT NULL, UNIQUE, DEFAULT). Support depends on the destination database.
   - *Alias*: an alternate display name for the field.
   - *Comment*: a description or note attached to the field.

   You can use buttons on the right side to:

- Add new field to add a new field to the destination table.
- Delete selected field to remove a field from the destination table.
- Move selected field up or Move selected field down to reorder fields.
- Reset all fields to clear all fields and start from scratch.
- Convert All Fields To Lowercase or Convert All Fields To Uppercase, to change the case of all field names.

5. Optionally, apply a spatial filter with *Filter by Extent*. This restricts the import to features within a defined bounding box.
6. Optionally, apply an attribute filter with *Feature filter*. This allows you to use an expression to select which features are imported.

4.2.7 Kaladėlės ir Žiniatinklio paslaugos

4.4 Table : Available actions for web services (1)

Level and Context menu	Paslaugų tipai				
	WMS / WMTS (*Informacija*)	XYZ Tiles (*Informacija*)	WCS (*Informacija*)	WFS / OGC API - Features (*Informacija*)	ArcGIS REST Servers (*Informacija*)
Top menu					
Kurti *Naują jungtį…*	☑	☑	☑	☑	☑
Įrašyti jungčių… detales į failą	☑	☑	☑	☑	☑
Įkelti jungtis…	☑	☑	☑	☑	☑
Connection					
Atnaujinti jungtį	☑	☑	☑	☑	☑
Keisti jungties… nustatymus	☑	☑	☑	☑	☑
Dubliuoti jungtį	☑	☑	☑	☑	☑
Išimti jungtį…	☑	☑	☑	☑	☑
Žiūrėti paslaugos informaciją naršyklėje					☑
Table / Layer					
Eksportuoti sluoksnį ► Į failą…	☑	☑	☑	☑	☑
Pridėti sluoksnį į projektą	☑	☑	☑	☑	☑
Add Filtered Layer to Project…					☑
Atverti *Sluoksnio savybių…* dialogą	☑	☑	☑	☑	☑
Žiūrėti paslaugos informaciją naršyklėje					☑

4.5 Table : Available actions for web services (2)

Level and Context menu	Paslaugų tipai				
	Vector Tiles (*Informacija*)	Scene (*Informacija*)	Cloud (*Informacija*)	SensorThings (*Informacija*)	STAC (*Reference*)
Top menu					
Kurti *Naują jungtį…*		☑	☑	☑	☑
Kurti *Naują bendrą jungtį…*	☑				
Kurti *Naują ArcGIS Vektorinių kaladėlių paslaugos jungtį…*	☑				
Save Connections… details to a file	☑	☑	☑	☑	☑
Įkelti jungtis…	☑	☑	☑	☑	☑
Connection					
Keisti jungties… nustatymus	☑	☑	☑	☑	☑
Dubliuoti jungtį	☑	☑	☑	☑	
Delete Connection…	☑	☑	☑	☑	☑
Refresh					☑
Details…					☑
Table / Layer					
Eksportuoti sluoksnį ► Į failą…				☑	
Pridėti sluoksnį į projektą	☑	☑		☑	
Atverti *Sluoksnio savybių…* dialogą	☑	☑		☑	
Refresh					☑
Download assets…					☑
Details…					☑

4.2.8 Projektas ir palaikomi failų tipai

4.6 Table : Available actions for various supported files :class: longtable

Failo tipas	Konteksto meniu
QGIS Projekto failas (.qgz, .qgs)	*Atverti projektą*
	Ištraukti simbolius… (atveria stilių tvarkyklę simbolių eksportavimui į XML, pridėjimui į numatytąjį stilių ar eksportavimui kaip PNG/SVG)
	Failo savybės…
QGIS sluoksnio apibrėžimas (.qlr)	*Eksportuoti sluoksnį ► Į failą…*
	Pridėti sluoksnį į projektą
	Sluoksnio savybės…
Apdorojimo modelis (.model3)	*Vykdyti modelį…*
	Keisti modelį…
Spausdinimo maketo šablonas (.qpt)	*Naujas išdėstymas pagal šabloną*
Python scenarijus (.py)	*Vykdyti scenarijų…*
	Atverti išoriniame redaktoriuje

SKYRIUS 5

QGIS konfigūracija

QGIS galima labai stipriai konfigūruoti. *Nustatymų* meniu teikia skirtingus įrankius, leidžiančius:

- *Stilių tvarkyklė…*: kurti ir tvarkyti *simbolius, stilius ir spalvų rampas*.
- *Savos projekcijos…*: kurti jūsų savas *koordinačių atskaitos sistemas*.
- *Klavišų kombinacijos…*: nustatyti jūsų pačių *klavišų kombinacijas*. Jas taipogi galite permušti kiekvienoje QGIS sesijoje, naudodami *projekto savybes* (randamas *Projekto* meniu).
- *Sąsajos tinkinimas…*: konfigūruoti *aplikacijos sąsają*, paslepiant dialogus ar įrankius, kurių jums nereikia.
- *Parinktys…*: nustatyti globalias *parinktis*, kurios taikomos įvairiose programinės įrangos srityse. Šios parinktys įrašomos į aktyvaus *Naudotojo profilio* nustatymus ir taikomi pagal nutylėjimą, kai tik su šiuo profiliu atidarote naują projektą.

5.1 Parinktys

Kai kurios bazinės QGIS parinktys gali būti keičiamos dialoge *Parinktys*. Parinkite meniu *Nustatymai* ► *Parinktys*. Jūs galite keisti parinktis pagal savo poreikius. Kai kuriems pakeitimams gali tekti iš naujo paleisti QGIS, kad jie įsigaliotų.

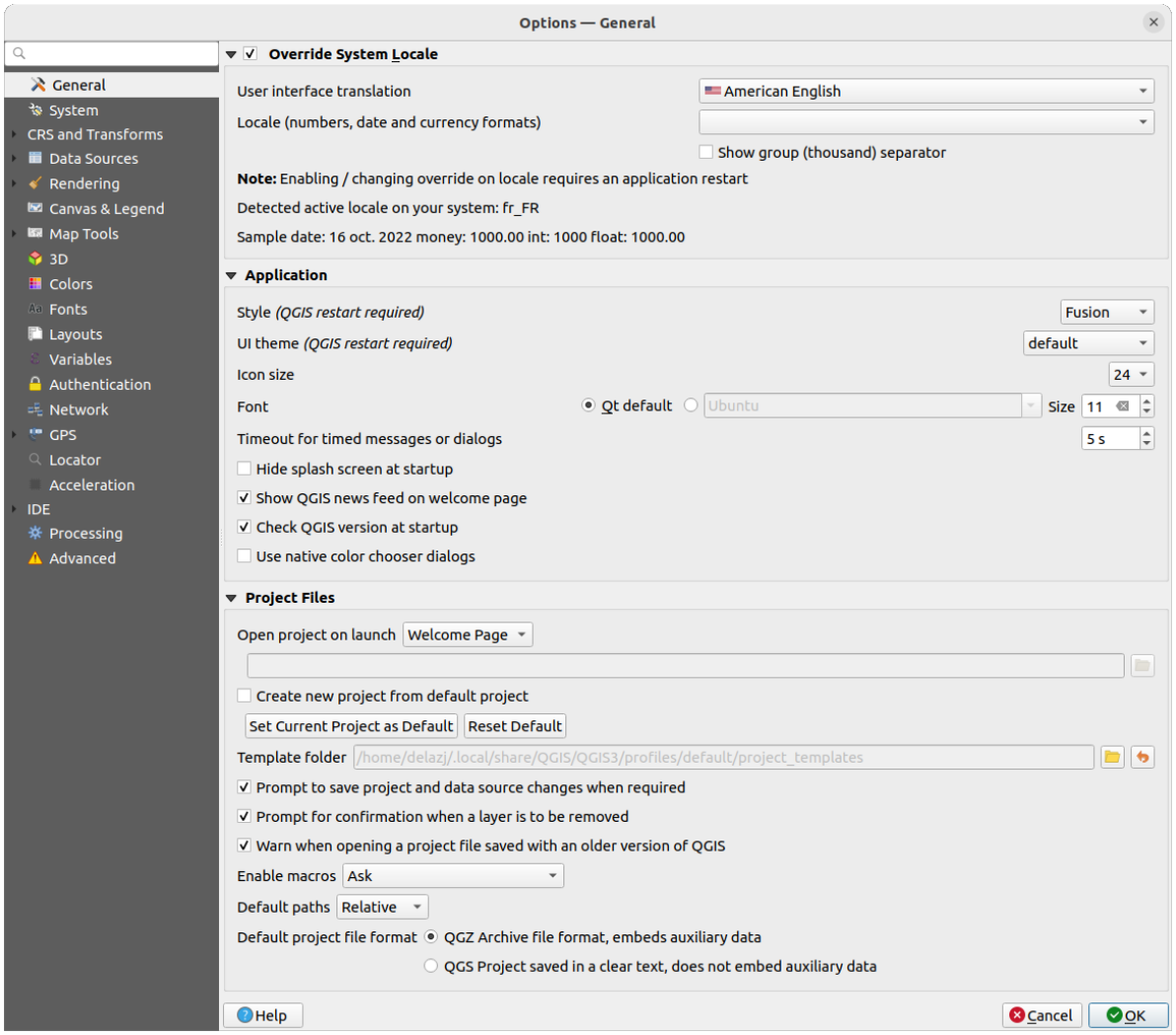
Žemiau aprašomos kortelės, kuriose galite tinkinti savo parinktis.

Pastaba

Priedai gali įtraukti savo nustatymus Parinkčių dialoge

Nors žemiau pateikti tik baziniai nustatymai, pastebėtina, kad šį sąrašą galima išplėsti *įdiegiant priedus*, kurie prideda savo parinktis į standartinį Parinkčių dialogą. Taip išvengiama situacijos, kai kiekvienas priedas turi savo atskirą konfigūravimo dialogą su papildomais meniu elementais, skirtais tik jiems…

5.1.1 Bendrieji nustatymai



5.1 Fig. : Bendrieji nustatymai

Perrašyti sistemos lokalę

Pagal nutylėjimą QGIS jūsų kalbą ir skaičių tvarkymą nustato pagal Operacinės sistemos konfigūraciją. Įjungus šią grupę jūs galite patys tinkinti šią elgseną.

- Parinkite jūsų GUI naudojamą *Naudotojo sąsajos vertimą*
- *Lokalė (skaičių, datų ir valiutų formatai)* nurodo, kaip rodyti skaitinių reikšmių ir datų duomenis
- *Rodyti grupės (tūkstančių) skirtuką*

Parinktų nustatymų suvestinę ir kaip jie bus interpretuojami matysite skilties apačioje.

Programa

- Pasirinkite *Stilius (reikia iš naujo paleisti QGIS)*, t.y. valdiklių išvaizdą ir išdėliojimą dialoguose. Galimos reikšmės priklauso nuo jūsų Operacinės sistemos.
- Nustatykite *Naudotojo sąsajos temą (reikia iš naujo paleisti QGIS)* . Galima rinktis iš „default“ (numatytoji), „Night Mapping“ (tamsi) arba „Blend of Gray“ (pilka)
- Parinkite *Piktogramų dydį*

- Nurodykite *Šriftą* ir jo *Dydį*. Šriftas gali būti *Qt numatytasis* arba nurodytas naudotojo
- Keiskite *Laikinų pranešimų ir dialogų rodymo laiką*
- *Paleidžiant nerodyti įvadinio paveikslo*
- *Rodyti QGIS naujienų kanalą pasveikinimo puslapyje*: rodo atrinktas QGIS naujienas, suteikiant jums galimybę sužinoti apie projekto naujienas (naudotojų/vystytojų susitikimų datas ir suvestines, bendruomenės apklausas, laidų pranešimus, įvairius patarimus…)
- *Tikrinti QGIS versiją paleidžiant*, kad gautumėte pranešimą, jei yra išleista naujesnė versija
- *Naudoti paprastus spalvų parinkimo dialogus* (žr. *Spalvų parinkiklis*)

Projektų failai

- *Paleidus atverti projektą*
  - „Pasveikinimo puslapis“ (numatytoji parinktis): gali rodyti „Naujienas“, projektų šabloną(us) ir paskutinius *naudotojo profilio* naudotus projektus (su miniatiūromis). Pagal nutylėjimą neatidaromas joks projektas.
  - „Naują“: atidaro naują projektą, pagal numatytąjį šabloną
  - „Paskutinį atvertą“: atidaro paskutinį jūsų įrašytą projektą
  - ir „Konkretų“: atveria konkretų projektą. Naudokite mygtuką …, kad nurodytumėte pagal nutylėjimą naudojamą projektą.
- *Kurti naują projektą iš numatytojo projekto*. Jūs turite galimybę spausti *Padaryti dabartinį projektą numatytuoju* arba *Atstatyti pradinį*. Galite naršyti po failus ir nurodyti aplanką, kuriame bus ieškoma jūsų projekto šablonų. Tai bus pridėta į *Projektas* ► *Naujas pagal šabloną*. Jei iš pradžių įjungsite *Kurti naują projektą iš numatytojo projekto* ir tada įrašysite projektą į šablonų aplanką.
- *Prireikus pasiūlyti įrašyti projekto ir duomenų šaltinio pakeitimus*, kad neprarastumėte jūsų padarytų pakeitimų.
- *Išimant sluoksnį prašyti patvirtinimo*
- *Įspėti, kai atidaromas projekto failas, sukurtas senesne QGIS versija*. Jūs visada galite atidaryti projektus, sukurtus senesne QGIS versija, bet įrašę tokį projektą, nebūtinai galėsite jį atidaryti su senesne laida, nes kai kurios savybės gali būti neprieinamos toje versijoje.
- *Įjungti projekto įtrauktą pythono kodą* . Ši parinktis valdo vykdymą makro, kurie vykdo veiksmus, atsitikus projekto įvykiams bei teiki savo pythono funkcijas kaip išraiškas. Jūs galite pasirinkti iš „Niekada“, „Paklausti“, „Tik šiai sesijai“ ir „Visada (nerekomenduojama)“.
- *Numatytasis kelių įrašymas*: nurodo, ar keliai iki failų ir sluoksnių įrašomi kaip „Absoliutūs“ ar „Santykiniai“ nuo projekto failo. Šį nustatymą galima permušti projekto lygyje.
- *Numatytas projekto failo formatas*
  - *QGZ failo archyvo formatas, įtraukia papildomus duomenis* (žr. *papildomi duomenys*)
  - *QGS projektas įrašomas atviru tekstu, neįtraukiant papildomų duomenų*: papildomi duomenys laikomi atskirame `.qgd` faile greta projekto failo.

5.1.2 Sistema

SVG keliai

Pridėkite ar išimkite *Kalius, kuriuose ieškoti vektorinių (SVG) simbolių*. Šie SVG failai vėliau prieinami simbolizacijai, etiketėms ar jūsų žemėlapio kompozicijos dekoravimui.

Taipogi skaitykite skyrių *Nutolusio arba įtraukto failo parinkiklis* skirtingiems būdams nurodyti svg failus QGIS kelyje.

Priedų keliai

Pridėkite ar išimkite *Kelius, kuriuose reikia ieškoti papildomų C++ priedų bibliotekų*.

Dokumentacijos keliai

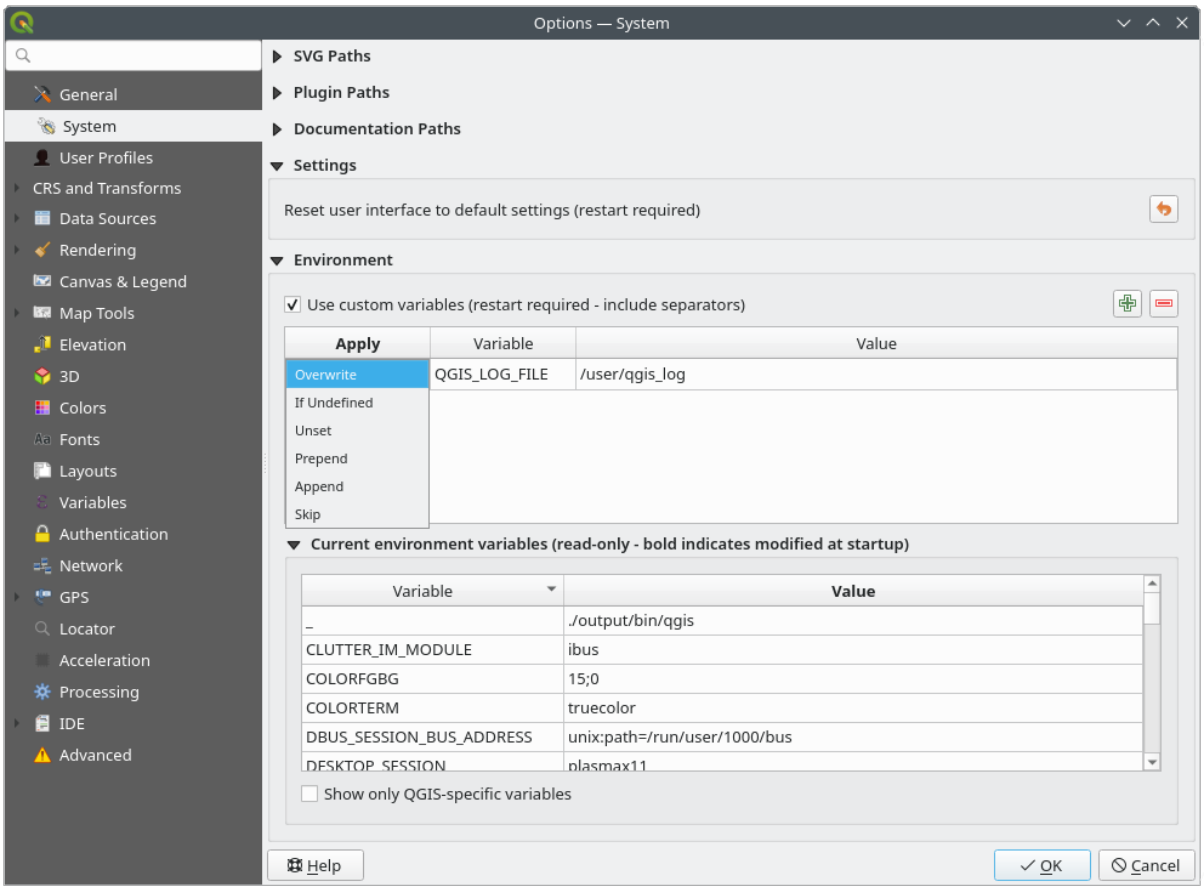
Pridėkite ar išimkite *Dokumentacijos kelius*, kurie naudojami QGIS pagalbai. Pagal nutylėjimą pridedama nuoroda į oficialią internetinę naudotojo dokumentaciją, atitinkančią naudojamą versiją (t.y., `https://docs.qgis.org/$qgis_short_version/$qgis_locale/docs/user_manual/`). Jūs galite pridėti kitas nuorodas ir prioritetizuoti jas nuo viršaus žemyn: kiekvieną kartą, kai dialoge paspaudžiate mygtuką *Pagalba*, parenkama aukščiausiai esanti nuorodą ir tikrinama, ar toks puslapis yra, jei ne - bandoma tolesnė nuoroda ir t.t..

> **Pastaba**
>
> Dokumentacijos versijos kuriamos ir verčiamos tik QGIS ilgo laikotarpio laidoms (LTR), tai reiškia, kad jei naudojate paprastą laidą (pvz. QGIS 3.0), pagalbos mygtukas pagal nutylėjimą atidarys kitos LTR vadovo puslapį (t.y. 3.4 LTR), kurioje gali būti aprašytas naujesnės laidos (3.2 ir 3.4) funkcionalumas. Jei nėra LTR dokumentacijos, tada bus naudojama „testavimo“ dokumentacija, su savybėmis iš naujesnių ir net vystymo versijų.

Nustatymai

Leidžia jums *Atstatyti naudotojo sąsajos numatytuosius nustatymus (reikės paleisti iš naujo)*, jei esate padarę kokius nors *tinkinimus*.

Aplinka



5.2 Fig. : Sisteminiai aplinkos kintamieji

System environment variables can be viewed, and many configured, in the **Environment** group. This is useful for platforms, such as Mac, where a GUI application does not necessarily inherit the user's shell environment. It's also useful for setting and viewing environment variables for the external tool sets controlled by the Processing toolbox (e.g., GRASS), and for turning on debugging output for specific sections of the source code.

Įjunkite varnelę *Naudoti savus kintamuosius (reikia paleisti iš naujo - su skirtukais)* ir jūs galėsite Pridėti ir Išimti aplinkos kintamuosius. Kiekvienam naujam elementui jūs galite sukonfigūruoti *Kintamojo* pavadinimą, jo *Reikšmę* ir naudojamą *Pritaikymo* būdą, tarp kurių yra:

- *Perrašyti*: pakeisti bet kokią esamą kintamojo reikšmę
- *Jei nenustatyta*: naudoti šią reikšme kintamajam, jei kintamasis dar neapibrėžtas aukštesniame lygmenyje (pvz. OS ar aplikacijos lygiuose)
- *Panaikinti*: panaikinti kintamąjį iš aplinkos (parametro *Reikšmė* nenaudojama)
- *Pridėti priekyje*: pridėti reikšmę jau esamo kintamojo priekyje
- *Pridėti gale*: pridėti reikšmę į esamo kintamojo pabaigą
- *Praleisti*: elementas paliekamas sąraše ateičiai, bet kol kas nenaudojamas

Jau apibrėžti aplinkos kintamieji rodomi skiltyje *Dabartiniai aplinkos kintamieji*, kur juos galima filtruoti įjungus varnelę *Rodyti tik su QGIS susijusius kintamuosius*.

5.1.3 Naudotojų profilių nustatymai

Pastaba

Daugiau informacijos apie tai, kaip valdyti naudotojų profilius, rasite tam skirtoje skiltyje *Darbas su naudotojo profiliais*.

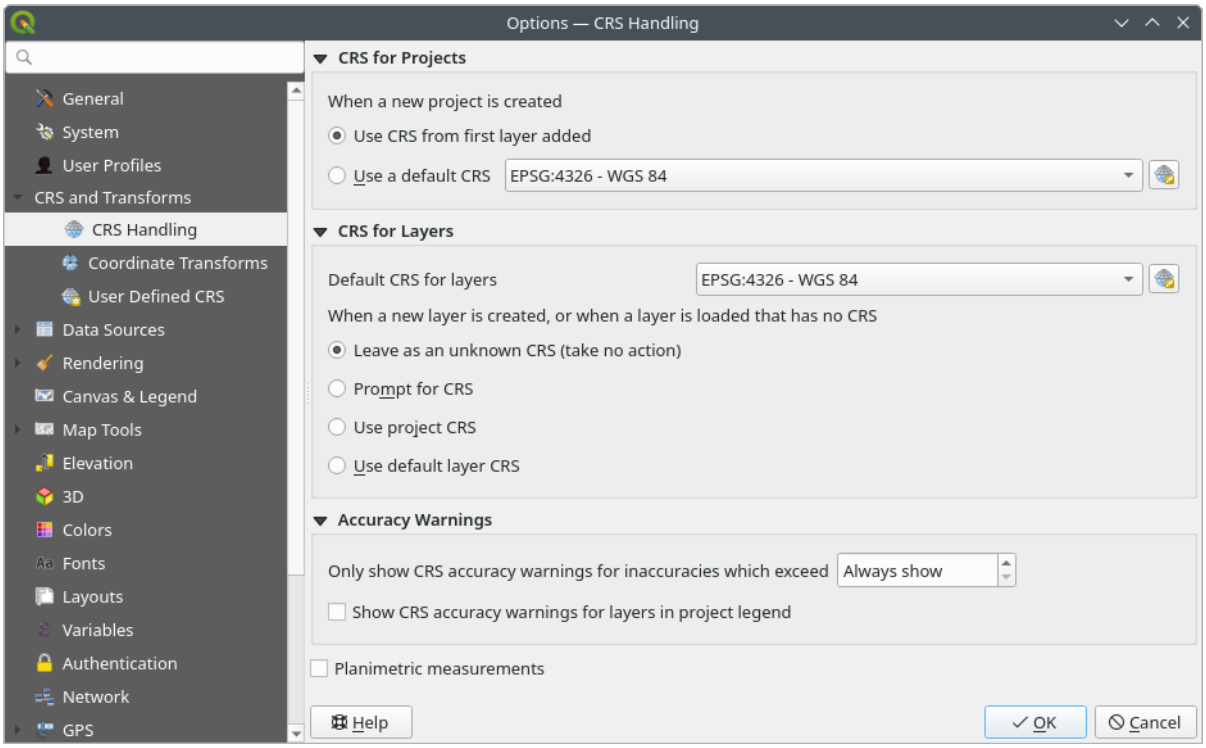
5.1.4 CRS ir transformavimo nustatymai

Pastaba

Daugiau informacijos apie tai, kaip QGIS tvarko sluoksnių projekcijas, rasite skyriuje *Darbas su projekcijomis*.

CRS valdymas

Kortelėje *CRS valdymas* jūs galite sukonfigūruoti CRS, kuri bus naudojama naujame projekte ar sluoksnyje.



5.3 Fig. : CRS nustatymai

CRS projektams

Yra parinktis automatiškai nustatyti naujo projekto CRS:

- *Naudoti CRS iš pirmo pridėto sluoksnio*: projekto CRS bus nustatyta tokia, kokia bus pirmame į projektą įkeltame sluoksnyje
- *Naudoti numatytąją CRS*: iš anksto nurodyta CRS pagal nutylėjimą priskiriama visiems naujiems projektams ir nekeičiama, pridedant į projektą sluoksnius.

Pasirinkimas bus įrašytas ir naudojamas vėlesnėse QGIS sesijose. Projekto koordinačių atskaitos sistemą galime permušti kortelėje *Projektas* ► *Savybės…* ► *CRS*.

CRS sluoksniams

Numatytoji sluoksnių CRS: parinkite numatytąją CRS, kurią reikia naudoti kuriant naują sluoksnį

Taipogi galite nurodyti veiksmą, kurį reikia vykdyti kuriant naują sluoksnį arba kai įkeliamas sluoksnis be nurodytos CRS.

- *Palikti kaip nežinomą CRS (nieko nedaryti)*
- *Klausti CRS*
- *Naudoti projekto CRS*
- *Naudoti numatytąją sluoksnių CRS*

Tikslumo įspėjimai

Only show CRS accuracy warnings for inaccuracies which exceed a given distance: occurs when you are explicitly creating or modifying a dataset and select a CRS based on a datum ensemble with lower accuracy. The default is to `Always show` the warning if any inaccuracy.

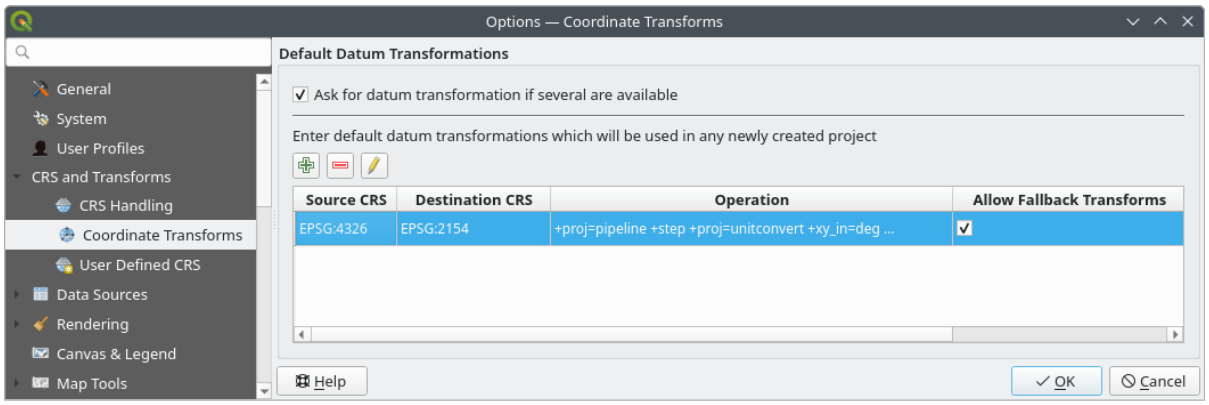
Rodyti CRS tikslumo įspėjimus sluoksniams projekto legendoje: įjungus šią parinktį, bet koks sluoksnis su CRS tikslumo problemomis (pvz. dinaminis crs be prieinamos koordinačių epochos arba CRS paremta duomenų ansamblius su netikslumu viršijančiu naudotojo nurodytą) turės įspėjimo piktogramą *Sluoksnių* skydelyje, rodantį, kad tai mažo tikslumo sluoksnis.

Tai sukurta naudojimui inžinerijoje, BIM, turto valdyme ir kitose srityse, kur metro/submetro netikslumai yra potencialiai labai pavojingi arba brangūs!

Planimetriniai matavimai: nurodo numatytąją savybės *planimetriniai matavimai* reikšmę naujai sukurtiems projektams.

Koordinačių transformavimas

Kortelė *Koordinačių transformacijos* padeda nurodyti koordinačių transformacijas ir veiksmus, vykdomus įkeliant sluoksnį į projektą ar perprojektuojant sluoksnį.



5.4 Fig. : Transformacijų nustatymai

Numatytosios datum transformacijos

Čia galite valdyti, ar perprojektuojant sluoksnius į kitą CRS reikia:

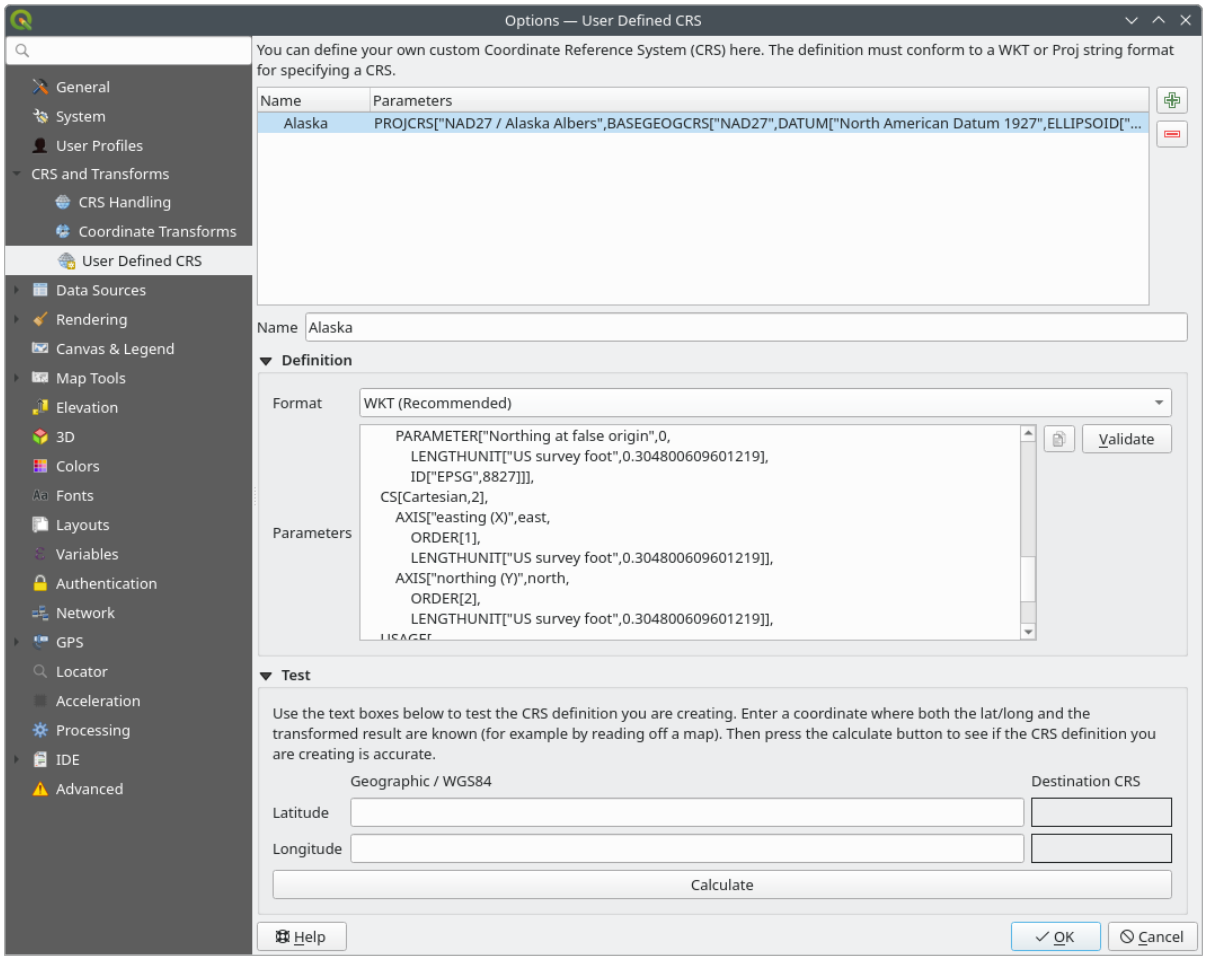
- automatiškai apdoroti naudojant QGIS numatytuosius transformacijos nustatymus;

- ir/arba papildomai pagal jūsų nurodymus:
  - *Klausti datum transformacijos, jei kelios yra galimos*
  - taikyti iš anksto nurodytą datum transformaciją. Daugiau informacijos rasite skyriuje *Duomenų transformacija*.

Jūs galite Pridėti, Išimti ar Keisti transformacijas, kurios bus naudojamos bet kuriame naujai sukurtame projekte.

Naudotojo apibrėžtos CRS

Kortelėje *Naudotojo apibrėžtos CRS* jūs galite apibrėžti savo CRS, kuri turi atitikti WKT arba Proj formatą.



5.5 Fig. : Naudotojo apibrėžtos CRS

Įrašykite *Pavadinimą* ir spauskite Pridėti naują CRS. Jei norite ištrinti jau esamą CRS, galite spausti Išimti CRS.

Apibrėžimas

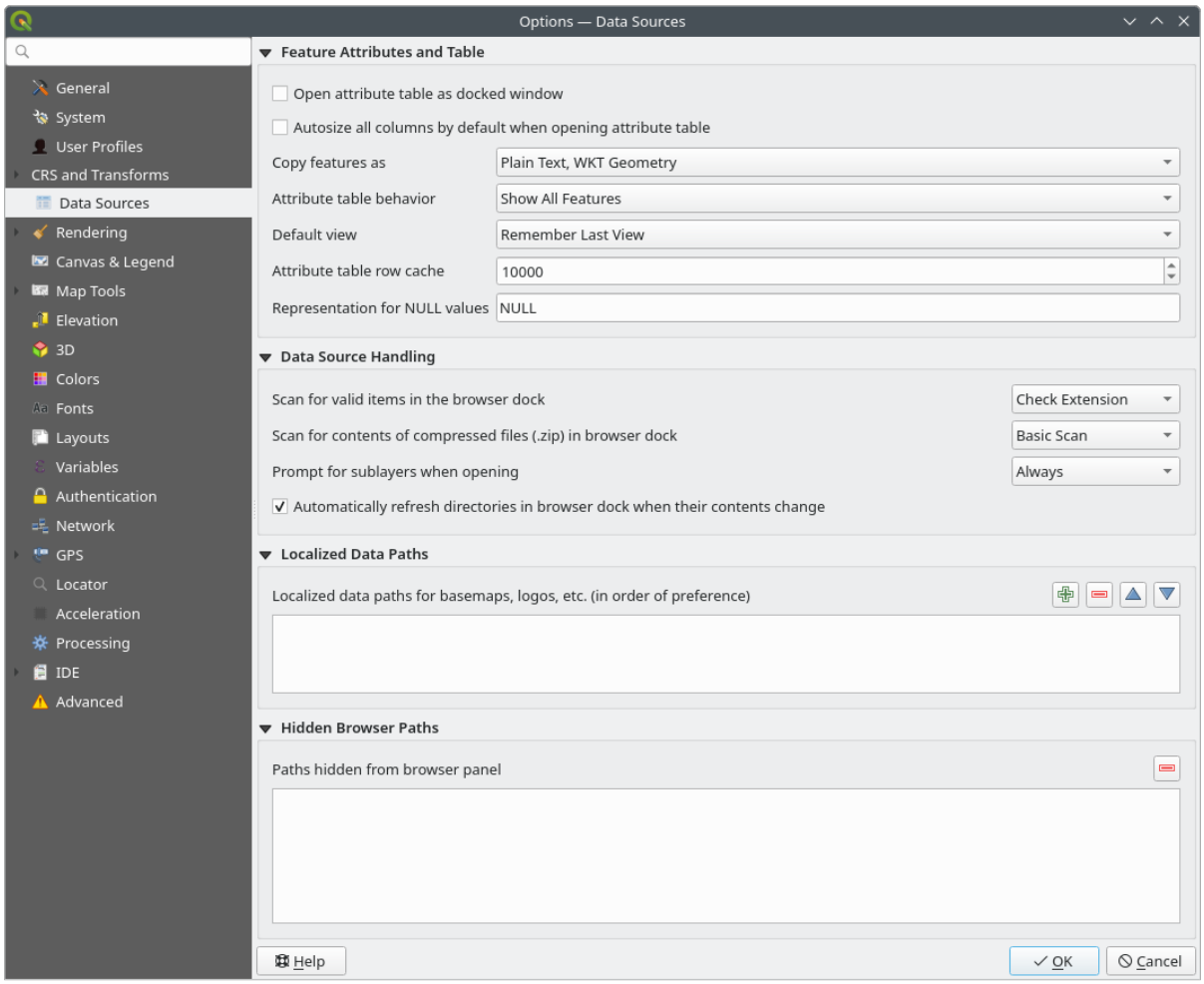
- ***Formatas***
  - WKT (Rekomenduojama)
  - Proj eilutė (senas - nerekomenduojama)
- ***Parametrai***

- Kopijuoti parametrus iš esamos CRS.
- *Tikrinti* patikrina, ar jūsų įvesta išraiška yra teisinga.

Tikrinti

Čia galite patikrinti jūsų sukurtą CRS apibrėžimą pagal Platumą ir Ilgumą. Naudokite žinomą koordinatę, kad pamatytumėte, ar jūsų apibrėžimas yra tikslus.

5.1.5 Duomenų šaltinių nustatymai



5.6 Fig. : Duomenų šaltinių nustatymai

Geoobjektų atributai ir lentelė

- *Atributų lentelę atverti prijungtame lange*
- *Automatiškai nustatyti visų stulpeliu plotį atidarant atributų lentelę*
- *Kopijuoti geoobjektus kaip*: apibrėžia geoobjektų formatą, kuris naudojamas duomenims, įkeliamiems kitose aplikacijose.
  - *Paprastas tekstas, be geometrijos*: atributai kopijuojami kaip tekstas, o geometrija praleidžiama
  - *Paprastas tekstas, WKT geometrija*: atributai kopijuojami kaip tekstas, o geometrija grąžinama kaip WKT

  - *Paprastas tekstas, WKB geometrija*: atributai kopijuojami kaip tekstas, o geometrija grąžinama kaip WKB (šešioliktainis formatas)
  - *GeoJSON*: atributai ir geoobjektas formatuojami kaip GeoJSON duomenys

- *Atributų lentelės elgsena* : atidarant nustatyti atributų lentelės filtrą. Yra trys galimybės: „Rodyti visus geoobjektus“, „Rodyti parinktus geoobjektus“ ir „Rodyti žemėlapyje matomus geoobjektus“.
- *Numatytasis vaizdas*: nurodykite atributų lentelės vaizdą kiekvieno atidarymo metu. Jis gali būti „Atsiminti paskutinį vaizdą“, „Lentelės vaizdas“ ar „Formos vaizdas“.
- *Atributų lentelės eilučių podėlis* . Šis eilučių podėlis leidžia įrašyti paskutines įkeltas N atributų eilučių, kad darbas su atributų lentele būtų greitesnis. Podėlis bus ištrintas uždarant atributų lentelę.
- *NULL reikšmių vaizdavimas*. Čia jūs galite nurodyti reikšmę, kuri bus rodoma laukams, turintiems NULL reikšmę.

Patarimas

Atributų lentelių su daug duomenų atidarymo pagerinimas

Dirbant su sluoksniais, kuriuose yra labai daug įrašų, atributų lentelės atidarymas gali būti lėtas, nes dialogas prašo ištraukti visas sluoksnio eilutes. Parinkus *Atributų lentelės elgseną* **Rodyti žemėlapyje matomus geoobjektus**, atidarant lentelę QGIS prašys tik geoobjektų, kurie matomi dabartinėje žemėlapio drobėje, dėl to duomenys bus įkelti daug greičiau.

Pastebėtina, kad atributų lentelėje esantys duomenys visada bus susieti su drobės apimtimi ją atidarius. Tai reiškia, kad tokioje lentelėje parinkus **Rodyti visus geoobjektus** daugiau geoobjektų parodyta nebus. Bet jūs galite atnaujinti rodomų geoobjektų aibę pakeisdami drobės apimtį ir parinkę **Rodyti žemėlapyje matomus geoobjektus** atributų lentelėje.

Duomenų šaltinio valdymas

- *Ieškoti tinkamų failų naršyklės skydelyje* . Galite parinkti „Tikrinti praplėtimą“ ir „Tikrinti failo turinį“.
- *Naršymo skydelyje tikrinti suspaustų failų (.zip) turinį* nurodo, kokio detalumo yra valdiklio informacija Naršyklės skydelio apačioje tikrinant tokius failus. Galimos parinktys yra „Netikrinti“, „Paprastas tikrinimas“ ir „Pilnas tikrinimas“.
- *Atidarant paprašyti posluoksnių*. Kai kurie rastrai palaiko posluoksnius — jie GDAL'e vadinami duomenų poaibiais. Tokie, pavyzdžiui, yra netCDF failai — jei yra daug netCDF kintamųjų, GDAL mato visus kintamuosius kaip poaibius. Ši parinktis leidžia jums valdyti, ką daryti su posluoksniais, kai atidaromas failas su posluoksniais. Jūs turite tokius pasirinkimus:
  - „Visada“: Visada prašyti (jei yra posluoksnių)
  - „Jei reikia“: Klausti, jei sluoksnis neturi juostų, bet turi posluoksnius
  - „Niekada“: Niekada neklausti, nieko neįkels
  - „Įkelti viską“: Niekada neklausti, visada įkelti visus posluoksnius
- *Automatiškai atnaujinti naršyklėje aplankus, kai jų turinys keičiasi*: Leidžia rankiniu būdu nebestebėti aplankų *Naršyklės* skydelyje (pvz. vengiant galimo sulėtėjimo dėl tinklo greičio).

Lokalizuoti duomenų keliai

Visiems failais paremtiems duomenų šaltiniams galima naudoti lokalizuotus kelius. Tai yra sąrašas kelių, kurie naudojami abstrahuojant duomenų šaltinio vietą. Pavyzdžiui, jei `C:\mano_žemėlapiai` yra įrašytas lokalizuotuose keliuose, sluoksnis su duomenų šaltiniu `C:\mano_žemėlapiai\mano_šalis\orto.tif` bus įrašytas projekte naudojant `localized:mano_šalis\orto.tif`.

Keliai rodomi prioriteto tvarka. T.y. QGIS visų pirma ieškos failo pirmame kelyje, tada antrame ir t.t.

Paslėpti naršyklės keliai

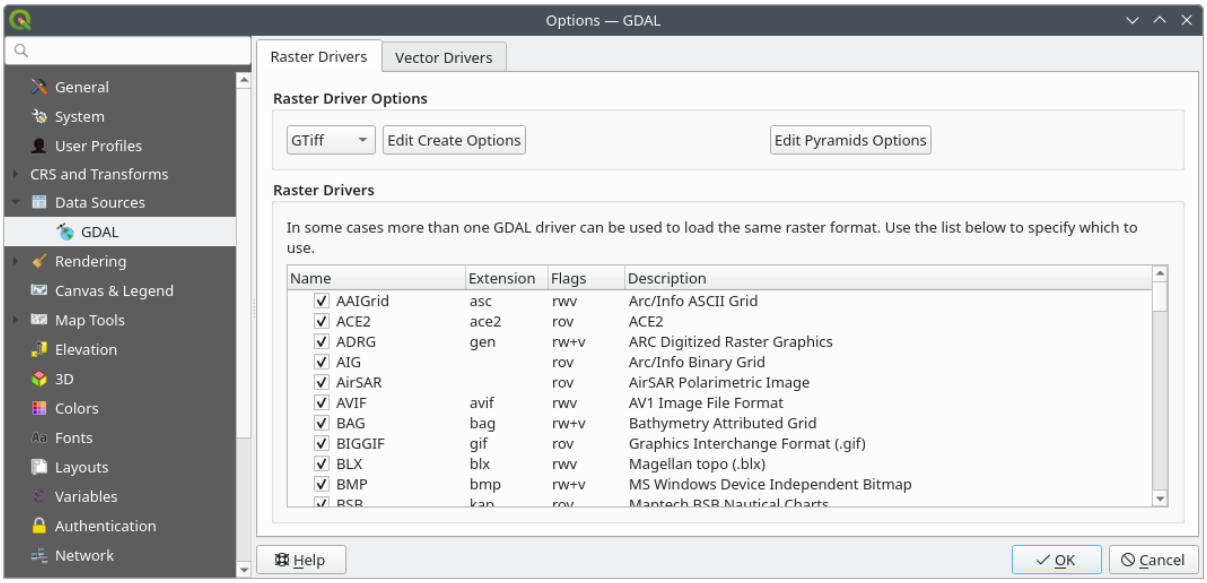
Šiame valdiklyje matysite visus aplankus, kuriuos jūs pasirinkote paslėpti iš *Naršyklės skydelio*. Pašalinus aplanką iš šio sąrašo jis vėl bus matomas *Naršyklės* skydelyje.

GDAL nustatymai

GDAL yra erdvinių duomenų apsikeitimo biblioteka, kuri palaiko didelį skaičių vektorinių ir rastro formatų. Ji teikia tvarkykles įvairių formatų duomenų skaitymui ir (dažnai) rašymui. *GDAL* kortelė demonstruoja šias rastro ir vektorių formatų tvarkyklių galimybes.

GDAL rastro ir vektoriaus tvarkyklės

Kortelės *Rastro tvarkyklės* ir *Vektorių tvarkyklės* leidžia nurodyti, kuri GDAL tvarkyklė naudojama rašyti ir/ar skaityti failus, nes kai kuriais atvejais gali būti prieinama daugiau nei viena GDAL tvarkyklė.



5.7 Fig. : GDAL nustatymai - rastro tvarkyklės

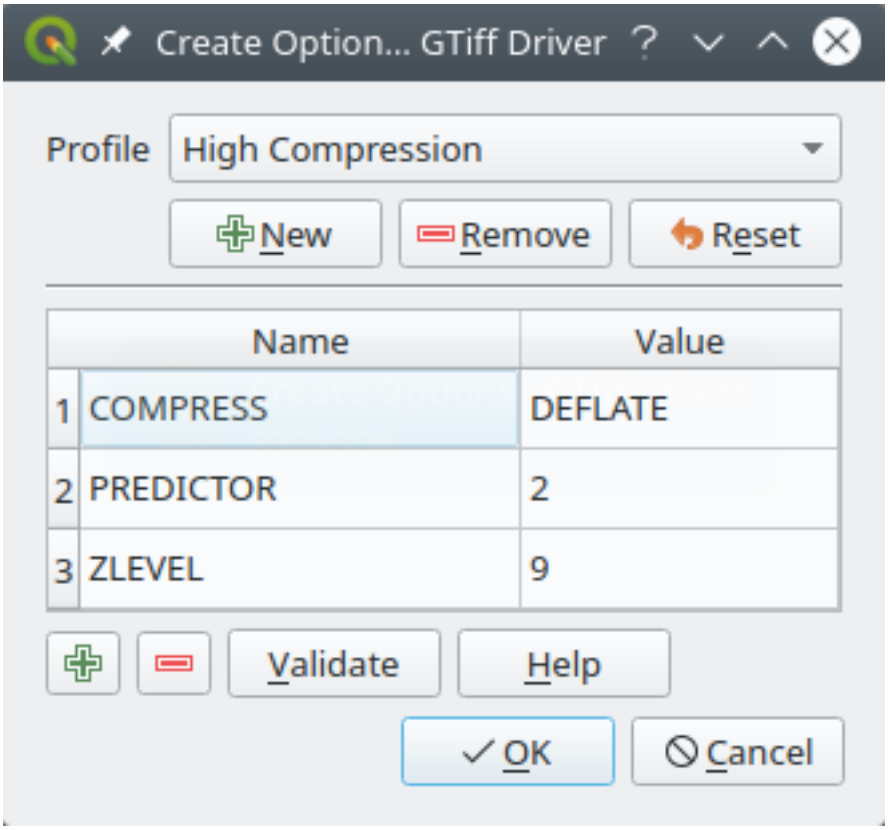
Patarimas

Du kartus paspaudus ant rastro tvarkyklės, kuri leidžia skaityti ir rašyti (`rw+ (v)`), atidaromas tinkinimo dialogas *Keisti kūrimo parinktis*.

Rastro tvarkyklių parinktys

Čia galite valdyti skaitymo ir rašymo prieigą palaikančių rastro tvarkyklių elgseną:

- *Edit create options*: allows you to edit or add different profiles of file transformation, i.e., a set of predefined combinations of parameters (type and level of compression, BIGTIFF support, blocks size, overview, colorimetry, alpha…) to use when outputting raster files. The parameters depend on the driver.



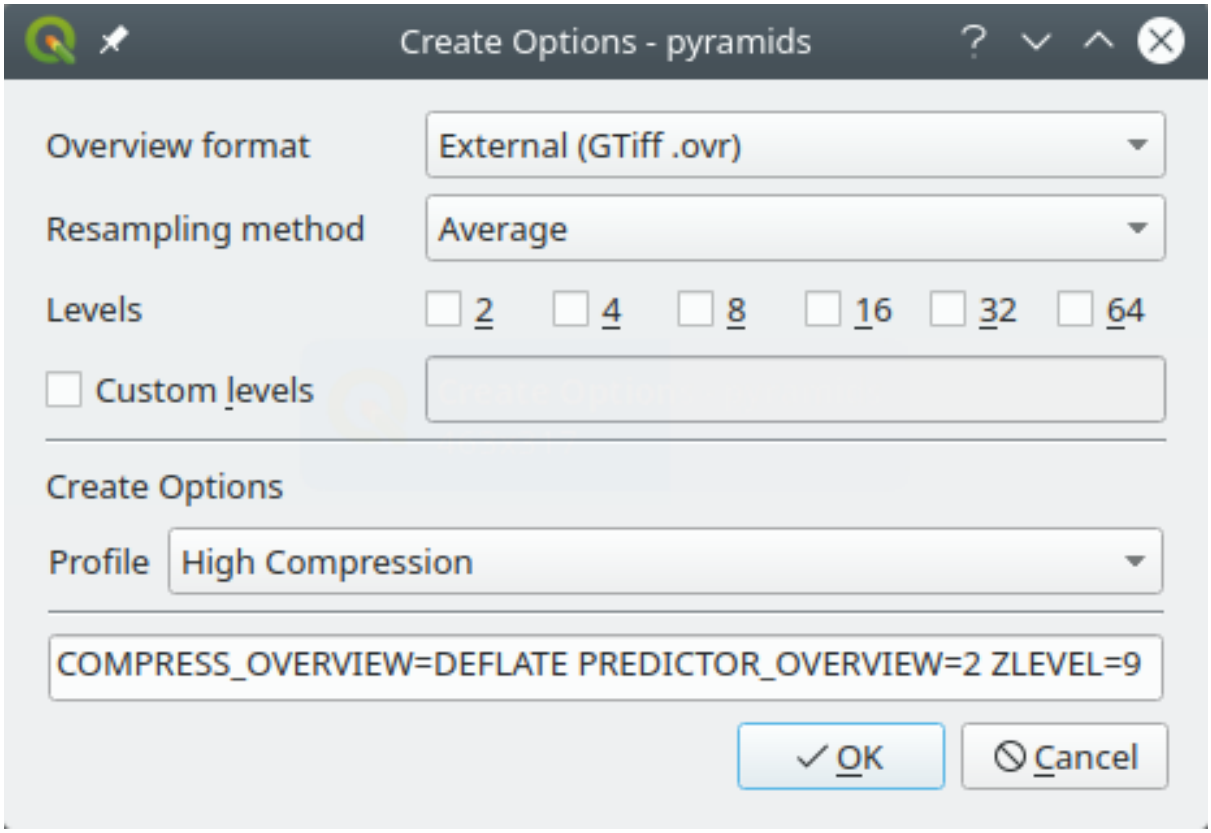
5.8 Fig. : Kūrimo parinkčių profilio pavyzdys (skirtas GeoTiff)

Viršutinėje dialogo dalyje yra esamų profilių sąrašas, kuriame jūs galite pridėti naujus ar išimti bet kurį iš esamų. Jūs taipogi galite atstatyti numatytuosius profilio parametrus, jei juos pakeitėte. Kai kurios tvarkyklės (pvz. GeoTiff) turi kelis pavyzdinius profilius, su kuriais galite dirbti.

Dialogo apačioje:

- Mygtukas leidžia pridėti eilutes, kuriose galite įrašyti parametro pavadinimą ir reikšmę
- Mygtukas ištrina pažymėtą parametrą
- Spauskite mygtuką *Tikrinti*, kad patikrintumėte, ar įvestos kūrimo parinktys tinka nurodytam formatui
- Naudokite mygtuką *Pagalba*, kad sužinotumėte, kokius galite naudoti parametrus, arba skaitykite GDAL rastro taisyklių dokumentaciją.

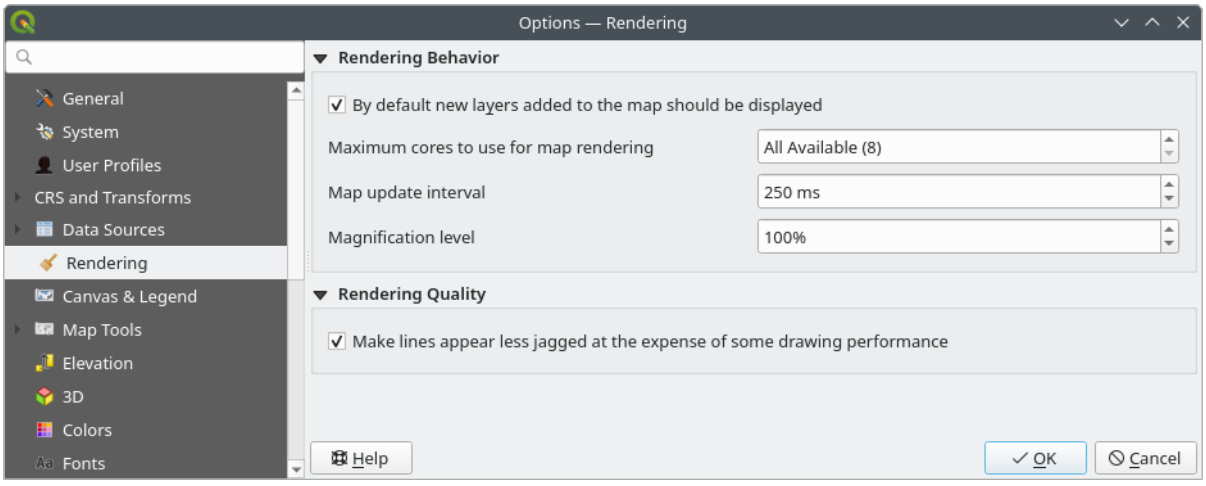
- *Keisti piramidžių parinktis*



5.9 Fig. : Piramidžių profilio pavyzdys

5.1.6 Braižymo nustatymai

Kortelėje *Braižymas* rasite sluoksnių braižymo žemėlapio drobėje nustatymus.



5.10 Fig. : Braižymo nustatymai

Braižymo elgsena

- *Naujai pridėti sluoksniai pagal nutylėjimą turi būti rodomi*: šią parinktį gali būti patogu išjungti įkeliant kelis sluoksnius, kad išvengtumėte kiekvieno sluoksnio braižymo tik jį pridėjus ir todėl bendro proceso sulėtinimo

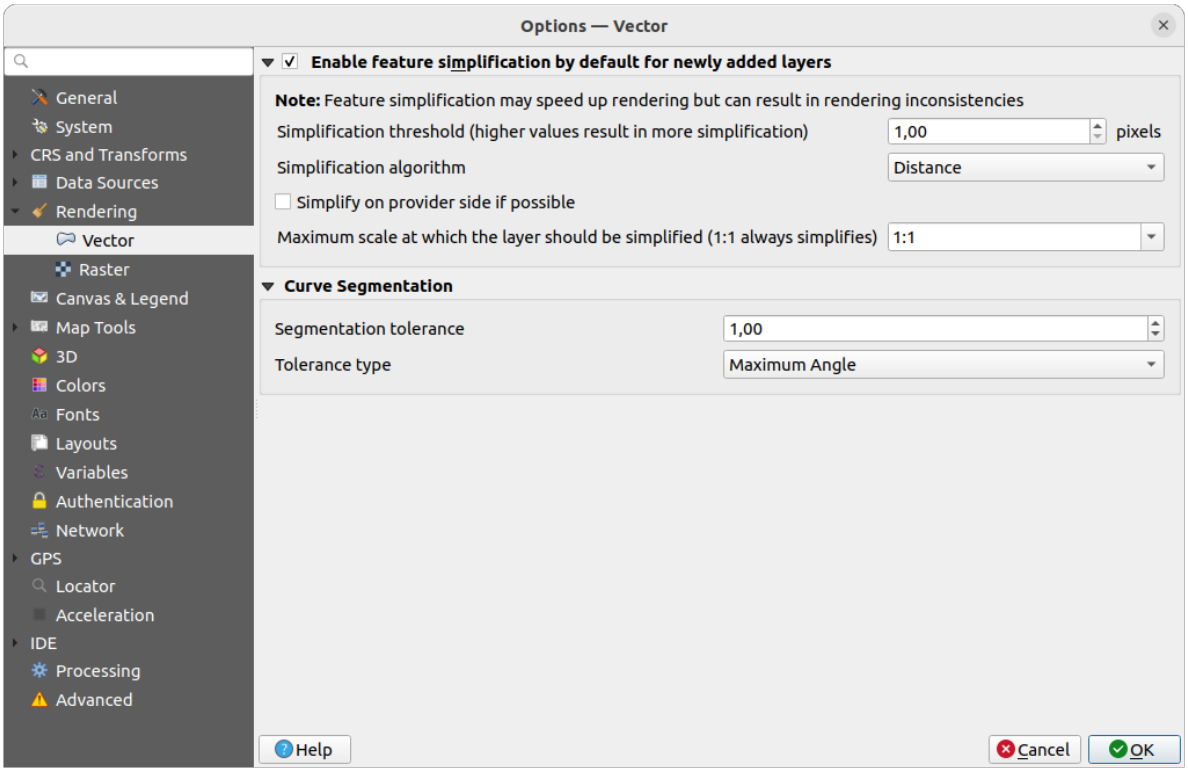
- Nustatykite parinktį *Maksimalus braižymui naudojamų branduolių skaičius*
- Žemėlapio drobė braižoma fone į atskirą piešinuką kas nurodytą *Žemėlapio atnaujinimo intervalą* (pagal nutylėjimą tai 250 ms), šio (nerodomo) piešinuko turinys bus panaudotas atnaujinant matomą reprezentaciją. Bet jei braižymas baigiamas greičiau, nei ši trukmė, rezultatas bus rodomas akimirksniu.
- *Didinimo lygis* (žr. *padidinimas*)

Braižymo kokybė

- ☑ *Braižyti mažiau kampuotas linijas, prarandant šiek tiek laiko*

Vektorių braižymo nustatymai

Kortelėje *Vektorius* rasite vektorinių sluoksnių braižymo nustatymus.



5.11 Fig. : Vektorių braižymo nustatymai

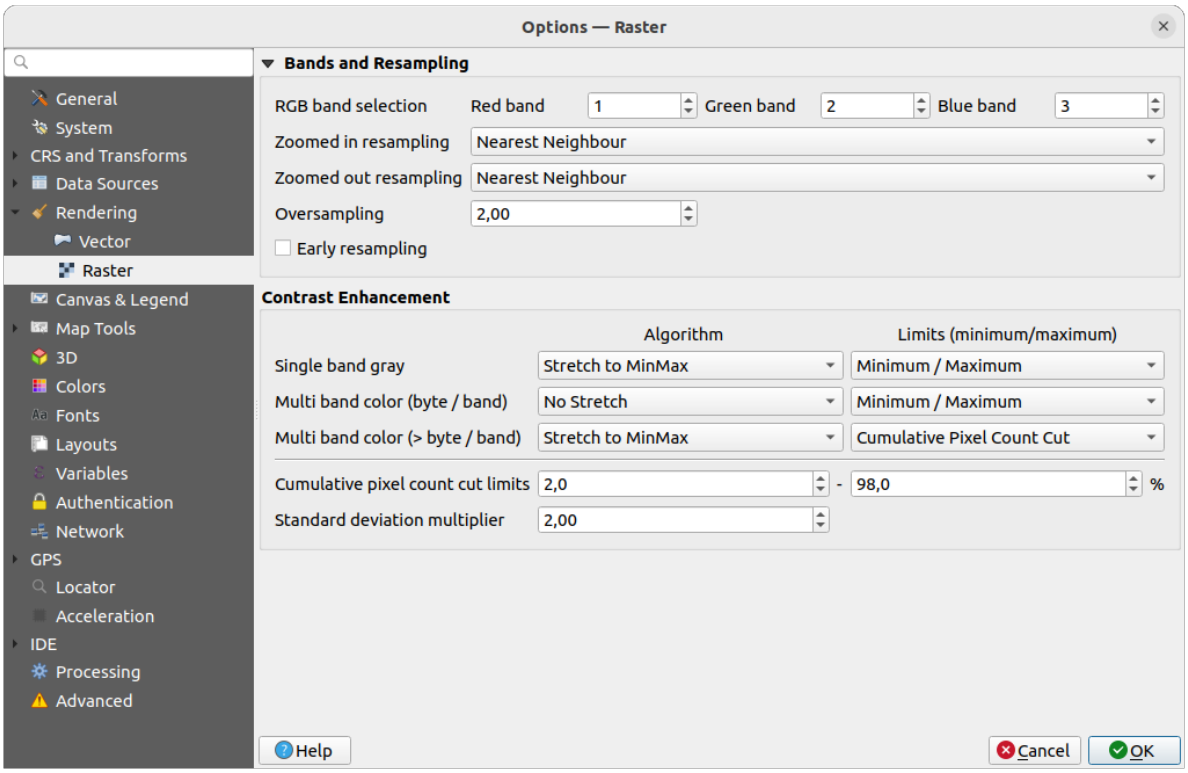
- ☑ *Pagal nutylėjimą įjungti geoobjektų supaprastinimą naujiems sluoksniams*: jūs paprastinate geoobjekto geometriją (paliekate mažiau viršūnių), todėl jie parodomi greičiau. Turėkite omenyje, kad dėl šito gali atsirasti braižymo nesutapimų. Galimos parinktys yra:
  - *Paprastinimo slenkstis (didesnės reikšmės reiškia daugiau paprastinimo)*
  - *Paprastinimo algoritmas*: Ši parinktis vykdo vietinį geometrijų paprastinimą, taip pagreitinant braižymą. Ji nekeičia iš duomenų teikėjo ištrauktos geometrijos. Tai svarbu, kai turite išraiškų, kurios naudoja geoobjekto geometrijas (pvz. skaičiuojate plotą) - taip užtikrinama, kad šie skaičiavimai atliekami su originaliomis geometrijomis, o ne su paprastintomis. Šiam tikslui QGIS turi tris algoritmus: „Atstumas“ (numatytasis), „PritrauktiPrieTinklelio“ ir „Visvalingam“.
  - ☐ *Simplify on provider side if possible*: the geometries are simplified by the provider (PostgreSQL, Oracle…) and unlike the local-side simplification, geometry-based calculations may be affected
  - *Didžiausias mastelis, kuriame sluoksnis turėtų būti paprastinamas (1:1 visada paprastina)*

> **Pastaba**
>
> Be globalių nustatymų, geoobjektų paprastinimą galima nustatyti bet kuriam konkrečiam sluoksniui per jo meniu *Sluoksnio savybės* ► *Braižymas*.

- *Kreivių segmentacija*
  - *Segmentacijos tolerancija*: šis nustatymas valdo, kaip braižomi lankai. **Kuo mažesnis** minimalus kampas (tarp dviejų greta esančių viršūnių ir kreivės centro laipsniais) ar didžiausias skirtumas (atstumas tarp dviejų viršūnių segmentų ir kreivės linijos žemėlapio vienetais), tuo **tiesesni** bus braižomi linijų segmentai.
  - *Tolerancijos tipas*: gali būti *Didžiausias kampas* arba *Didžiausias skirtumas* tarp aproksimacijos ir kreivės.

Rastro braižymo nustatymai

Kortelėje *Rastras* rasite rastro sluoksnių braižymo nustatymus.



5.12 Fig. : Rastro braižymo nustatymai

Skiltyje *Juostos ir perrinkimas*:

- Naudodami *RGB juostų parinkimą*, jūs galite nurodyti Raudonos, Žalios ir Mėlynos juostų numerius.
- Galite nurodyti metodus *Atrinkimą pritraukus* ir *Atrinkimą atitraukus*. *Atrinkimui pritraukus* jūs galite pasirinkti vieną iš trijų perrinkimo metodų: „Artimiausio kaimyno“, „Dvilinijinį (2x2 branduolys)“ ir „Kubinį (4x4 branduolys)“. *Atrinkimui atitraukus* jūs galite pasirinkti iš „Artimiausio kaimyno“ ir „Vidurkio“. Taipogi galite nurodyti *Per didelio pritraukimo* reikšmę (tarp 0.0 ir 99.99 - didelė reikšmė reiškia daugiau darbo QGIS'ui - numatytoji reikšmė yra 2.0).
- *Ankstyvas perrinkimas*: leidžia paskaičiuoti rastro braižymą tiekėjo lygmenyje, kur žinoma pradinė raiška, tai leidžia pasiekti geresnį mastelio braižymą naudojant QGIS savo stilius. Tikrai patogu kaladėlių rastrams,

įkeltiems naudojant *interpoliacijos metodą*. Parinktį taipogi galima įjungti ir sluoksnio lygmenyje (*Simbologijos* savybės)

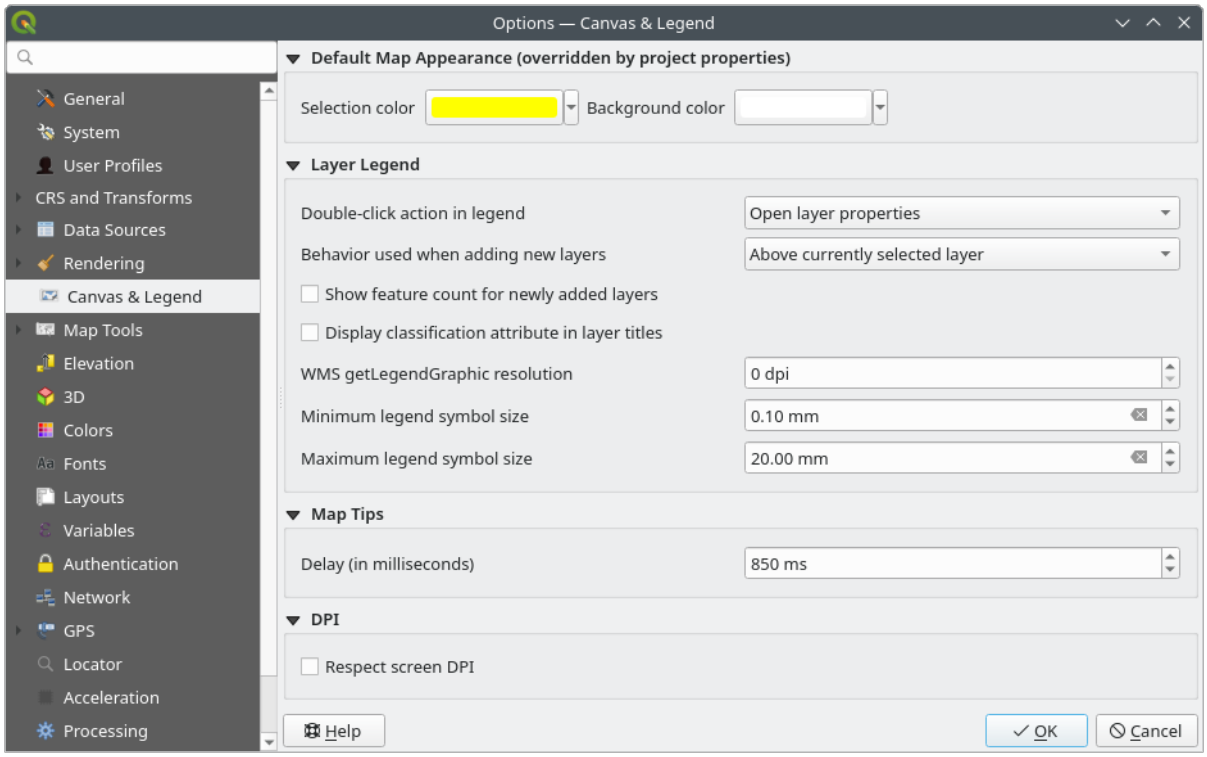
Kontrasto stiprinimo parinktys gali būti taikomos *Vienai juodai-baltai juostai*, *Kelių juostų spalvoms (baitas / juosta)* ar *Kelių juostų spalva (>baitas / juosta)*. Kiekvienam jūs galite nurodyti:

- Naudojamą *Algoritmą*, kurio reikšmės gali būti „Jokio ištempimo“, „Ištempti iki MinMax“, „Ištempti apkarpyti iki MinMax“ ar „Apkarpyti iki MinMax“
- Taikomi *ribojimai (minimumas/maksimumas)*, su reikšmėmis, tokiomis kaip „Kumuliacinė taškų skaičiaus riba“, „Minimumas/Maksimumas“, „Vidurkis +/- standartinis nuokrypis“.

Kontrasto stiprinimo parinktyse taipogi yra:

- *Kumuliacinė taškų skaičiaus riba*
- *Standartinio nuokrypio daugiklis*

5.1.7 Drobės ir legendos nustatymai



5.13 Fig. : Drobės ir legendos nustatymai

Šios savybės leidžia jums nurodyti:

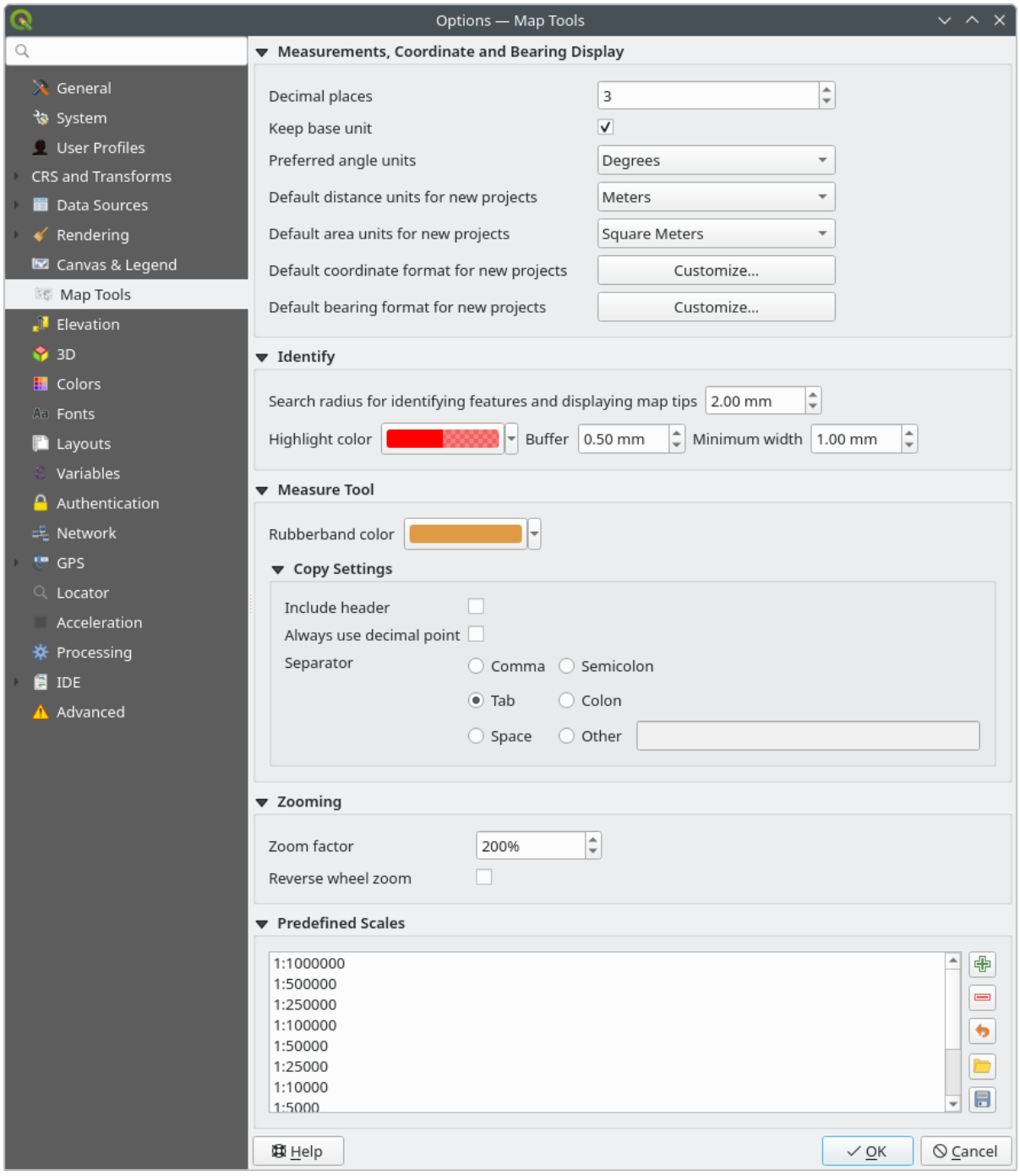
- **Numatytąją žemėlapio išvaizdą (projekto savybės ją permuša)**: *Parinkimo spalvą* ir *Fono spalvą*.
- **Sluoksnių legendos** veiksmai:
  - *Dvigubo paspaudimo veiksmas legendoje*: kai du kartus paspaudžiamas sluoksnis, ar turi būti *Atidaromos sluoksnio savybės* (numatytoji parinktis), *Atidaroma atributų lentelė* ar *Atidaromas sluoksnio stiliaus elementas*.
  - *Elgsena pridedant naujus sluoksnius*: nustato, kur *Sluoksnių* skydelyje pridedami sluoksniai, kai jie įkeliami į QGIS. Tai gali būti:
    - *Virš dabar parinkto sluoksnio*

        * *Visada sluoksnių medžio viršuje*
        * *Optimalus indeksas dabartinėje sluoksnių medžio grupėje*: Priešingai nei kitos dvi parinktys, kurios rikiuoja naujus sluoksnius ir padeda juos kaip krūvą norimoje pozicijoje, ši parinktis išplečia rikiavimo logiką į visą sluoksnių medį (arba aktyvią grupę) ir įterpia naujus sluoksnius „optimaliu“ būdu užtikrinant, kad taškų sluoksniai yra virš taškų sluoksnių, po kurių eina linijų sluoksniai virš linijų ir tada poligonų sluoksniai ir t.t.
    - ☐ *Rodyti geoobjektų skaičių naujai pridėtiems sluoksniams*: skydelyje *Sluoksniai* greta sluoksnio pavadinimo rodo geoobjektų skaičių. Jei yra, rodomas ir geoobjektų klasių skaičius. Galite ant sluoksnio spausti dešinį pelės mygtuką, jei norite įjungti/išjungti geoobjektų skaičiavimą.
    - ☐ *Rodyti klasifikavimo atributą prie sluoksnių pavadinimų*, pavyzdžiui kai taikomas kategorizuotas ar nuo taisyklių priklausantis simbolizavimas (daugiau informacijos rasite skyriuje *Symbology Properties*).
    - *WMS getLegendGraphic rezoliucija*
    - *Minimalus* ir *Maksimalus sutartinių ženklų simbolio dydis* valdo *Sluoksnių* skydelyje rodomo simbolio dydį
- Sluoksnių *žemėlapių patarimų* rodymo *Delsimas* milisekundėmis
- Ar QGIS turi ☑ *Atsižvelgti į ekrano DPI*: Įjungus šią parinktį, QGIS pasistengs rodyti drobę fiziškai teisingu masteliu ekrane, priklausomai nuo monitoriaus fizinio DPI. Simbologija su nurodytu rodymo dydžiu taipogi bus teisingai braižoma, pvz. 10mm simbolis bus ekrane rodomas 10mm dydžio. Bet etikečių šriftų dydžiai drobėje gali skirtis nuo QGIS naudotojo sąsajos ir kitų aplikacijų. Išjungus šį nustatymą QGIS naudos operacinės sistemos loginį DPI, kuris atitiks naudojamą kitose aplikacijose. Bet drobės mastelis ekrane gali būti fiziškai netikslus. Ypač tai jausis didelės raiškos ekranuose, kur simbologija tikėtina atrodys per smulki.

  Geriausiai patirčiai rekomenduojama įjungti ☑ *Atsižvelgti į ekrano DPI*, ypač kai naudojami keli arba skirtingi vaizduokliai ir ruošiami vizualiai aukštos kokybės žemėlapiai. Išjungus ☑ *Atsižvelgti į ekrano DPI* bus kuriama išvestis, labiau tinkama ekranui skirtus žemėlapius, ypač kur šriftų dydžiai atitiks naudojamus kitose aplikacijose.

> **Pastaba**
>
> Braižymas išdėstymuose neįtakojamas nustatymo *Atsižvelgti į ekrano DPI*; jie visada atsižvelgia į nurodytą išvesties įrenginio DPI. Taipogi atkreipkite dėmesį, kad šis nustatymas naudoja fizinį ekrano DPI tokį, kokį jį praneša operacinė sistema, o tai nebūtinai bus tikslu visiems vaizduokliams.

5.1.8 Žemėlapio įrankių nustatymai



5.14 Fig. : Žemėlapio įrankių nustatymai

Measurements, Coordinate and Bearing Display

This section provides ways to configure default units parameters:

- Nurodykite *Skaitmenų po kablelio* skaičių
- *Išlaikyti bazinį vienetą* automatiniam didelių skaičių konvertavimui (pvz. metrų į kilometrus)
- *Pageidaujami kampų vienetai*: galimos reikšmės yra „Laipsniai“, „Radianai“, „Gonai“, „Minutės“, „Sekundės“, „Sūkiai“, „Miliradianai (Si apibrėžimas)“ ar „Mil (NATO/karinis apibrėžimas)“

- *Default distance units for new projects*: options are ‚Meters', ‚Kilometers', ‚Feet', ‚Yards', ‚Miles', ‚Nautical Miles', ‚Centimeters', ‚Millimeters', ‚Inches', ‚Degrees' or ‚Map Units' (automatically matches the project's CRS units)
- *Preferred area units for new projects*: options are ‚Square meters', ‚Square kilometers', ‚Square feet', ‚Square yards', ‚Square miles', ‚Hectares', ‚Acres', ‚Square nautical miles', ‚Square centimeters', ‚Square millimeters', ‚Square inches', ‚Square degrees' or ‚Map Units' (automatically matches the project's CRS units)
- *Numatytasis koordinačių formatas naujiems projektams*, rodomas *Koordinačių* langelyje QGIS būsenos juostoje ir Identifikuoti geoobjektus įrankio rezultatų skiltyje *Išvestinė*
- *Numatytasis krypties formatas naujiems projektams*, būsenos juostoje rodomam drobės tempimui ir įrankio Matuoti kryptį.

The global options can be overridden at the project level, *here* for measurement units and *here* for coordinate and bearing formatting.

Identify tool

This group offers some options regarding the behavior of the *Identify tool*.

- *Geoobjekto identifikavimo ir žemėlapio patarimų spindulys* - tai tolerancijos atstumas - iki tokio atstumo paspaudus ant objekto, identifikavimo įrankis rodys rezultatus.
- *Paryškinimo spalva* leidžia jums parinkti, kokia spalva paryškinami identifikuojami objektai.
- *Buferis* nurodo identifikavimo paryškinimo braižymo buferį.
- *Minimalus plotas* nurodo kokio storio turi būti pažymėjimo kontūras.

Matavimo įrankis

- Nurodykite matavimo įrankių *Gumelės spalvą*

Matavimo įrankio kopijavimo nustatymai

Šie nustatymai leidžia valdyti lango Matuoti liniją mygtuko *Kopijuoti* elgseną. Galite pasirinkti *Įtraukti antraštę*, kad paliktumėte stulpelių pavadinimus, ar savo pasirinktą *Skirtuką*. Taipogi galite nurodyti *Visada naudoti dešimtainį tašką*, kad į iškarpinę nukopijuotose koordinatėse trupmeninė dalis būtų atskirta tašku, net jei jūsų kalbos QGIS parinkčių nustatymai nurodo naudoti kablelį. Šiuo atveju, jei neparinksite *Visada naudoti dešimtainį tašką*, negalėsite kablelio parinkti kaip laukų skirtuko.

Slinkimas ir didinimas

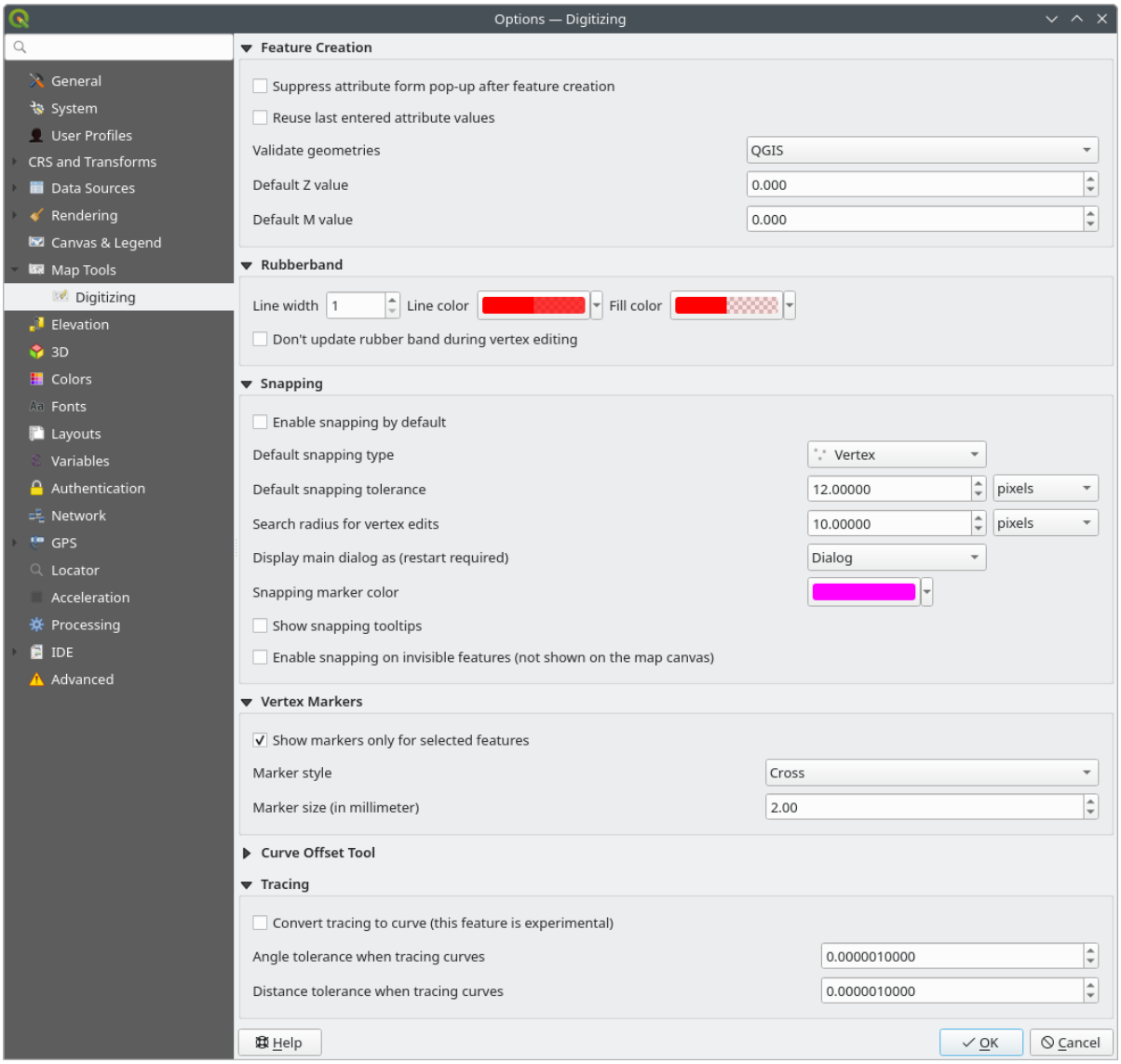
- *Didinimo daugiklis* mastelio keitimo įrankiams ar pelės ratukui
- *Atvirkštinis mastelio keitimas ratuku* leidžia jums keisti pelės ratuko slinkimo kryptį.

Numatyti masteliai

Čia rasite sąrašą numatytųjų mastelių, kurie pagal nutylėjimą rodomi su masteliu susijusiuose iškrentančių sąrašų valdikliuose, tokiuose kaip būsenos juostos *Mastelis*, matomumo mastelių parinkime arba antriniuose 2D žemėlapio vaizdo nustatymuose,… Su ir mygtukais jūs galite pridėti ir išimti jums reikiamus mastelius. Taipogi galite importuoti ar eksportuoti mastelius iš/į `.XML` failą. Pastebėtina, kad vis dar turite galimybę išimti jūsų pakeitimus ir atstatyti pradinį sąrašą.

Projekto savybių dialoge jūs taipogi galite nustatyti savo mastelių sąrašą, permušdami valdikliuose šį globalųjį.

Skaitmeninimo nustatymai



5.15 Fig. : Skaitmeninimo nustatymai

Ši kortelė leidžia jums konfigūruoti bendruosius *vektorinio sluoksnio redagavimo* (atributų ir geometrijos) nustatymus.

Geoobjektų kūrimas

- *Nerodyti atributų formos po geoobjekto sukūrimo*: šią parinktį galima permušti kiekvieno sluoksnio savybių dialoge.
- *Perpanaudoti paskutines įvestas atributų reikšmes*: atsiminti paskutines naudotas kiekvieno atributo reikšmes ir naudoti jas kaip pradines kitame vektorizuotame geoobjekte. Veikia sluoksnio lygmenyje. Šią elgseną taipogi galima valdyti lauko lygyje (žiūrėkite *Configure the field behavior*).
- *Validate geometries*. Editing complex lines and polygons with many nodes can result in very slow rendering. This is because the default validation procedures in QGIS can take a lot of time. To speed up rendering, it is possible to select GEOS geometry validation or to switch it off. GEOS geometry validation is much faster, but the disadvantage is that only the first geometry problem will be reported.

  Pastebėtina, kad priklausomai nuo pažymėjimo, geometrijos klaidų ataskaitos gali skirtis (žr. *Klaidų pranešimų tipai ir jų reikšmės*)

- *Numatytoji Z reikšmė* kuriant naujus 3D geoobjektus.

Gumelė

- Nurodykite gumelės *Linijos plotį*, *Linijos spalvą* ir *Pildymo spalvą*.
- *Nekeisti gumelės redaguojant viršūnes*.

Pritraukimas

- *Pagal nutylėjimą įjungti pritraukimą* įjungia pritraukimą atidarius projektą
- Nustatykite *Numatytąjį pritraukimo tipą* („Viršūnė“, „Segmentas“, „Centroidas“, „Segmento vidurys“, „Linijos pabaiga“, „Plotas“)
- Nustato *numatytąją pritraukimo toleranciją* žemėlapio vienetais ar taškais
- Nustato *Viršūnių keitimo paieškos spindulį* žemėlapio vienetais ar taškais
- *Pagrindinį dialogą rodyti kaip (reikės paleisti iš naujo)*: nurodykite, ar sudėtingesnį pritraukimo dialogą rodyti kaip „Dialogą“ ar „Prijungtą langą“.
- *Pritraukimo žymeklio spalva*
- *Rodyti pritraukimo patarimus*, tokius kaip pavadinimą sluoksnio, kurio geoobjektą jūs pritrauksite. Naudinga, kai persidengia keli geoobjektai.
- *Įgalinti pritraukimą prie nematomų (nerodomų žemėlapio drobėje) geoobjektų*

Viršūnių žymėjimas

- *Rodyti žymes tik pažymėtiems geoobjektams*
- Nustatykite viršūnės *Žymės stilių* („Kryžius“ (numatytas), „Pusiau permatomas apskritimas“ ar „Jokio“)
- Nustatykite viršūnės *Žymės dydį (milimetrais)*

Kreivės postūmio įrankis

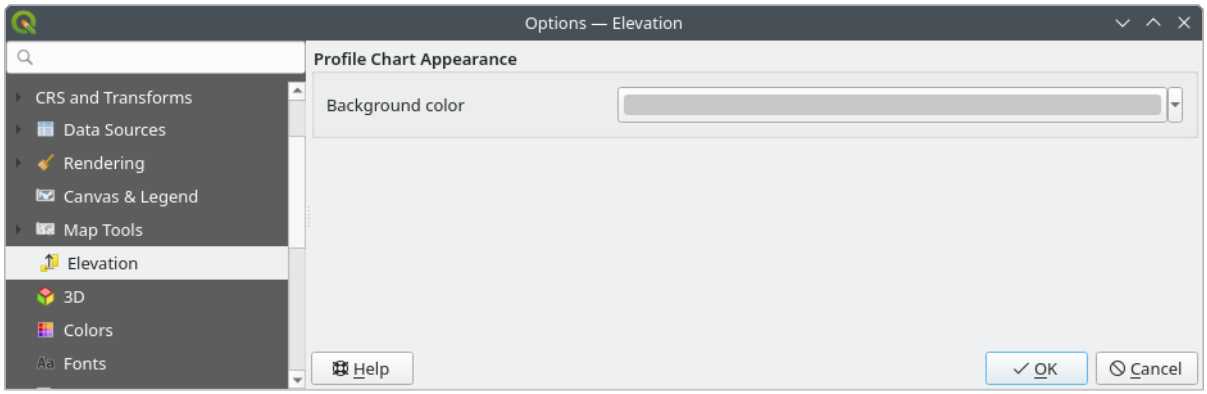
Kitos 3 parinktys skirtos įrankius Kreivės postūmio įrankis skyriuje *Advanced digitizing*. Naudojant kelis nustatymus galima įtakoti linijos pastūmimo formą.

- *Sujungimo stilius*: nurodo, kurie jungimai turi būti naudojami pastumiant kampus „Apvalūs“, „Mitra“ ar „Kampuoti“
- *Kvadranto segmentai*: valdo kiek linijos segmentų turi būti naudojami aproksimuoti ketvirtį apskritimo, kuriant apvalius poslinkius
- *Mitros riba*: Nustato maksimalų naudojamą atstumą nuo paslinktos geometrijos, kai kuriamas mitros sujungimas, kaip santyki atstumo atstumo (taikoma tik su mitros sujungimo stiliumi)
- *Pabaigos stilius*: valdo kaip tvarkomos linijų pabaigos, jos gali būti „Apvalios“, „Plokščios“ ar „Kvadratinės“.

Braižymas

Įjungus *Keisti braižymą į kreivę* jūs skaitmenizuojant galite kurti kreivių segmentus. Atsiminkite, kad jūsų duomenų tiekėjas turi palaikyti šią savybę.

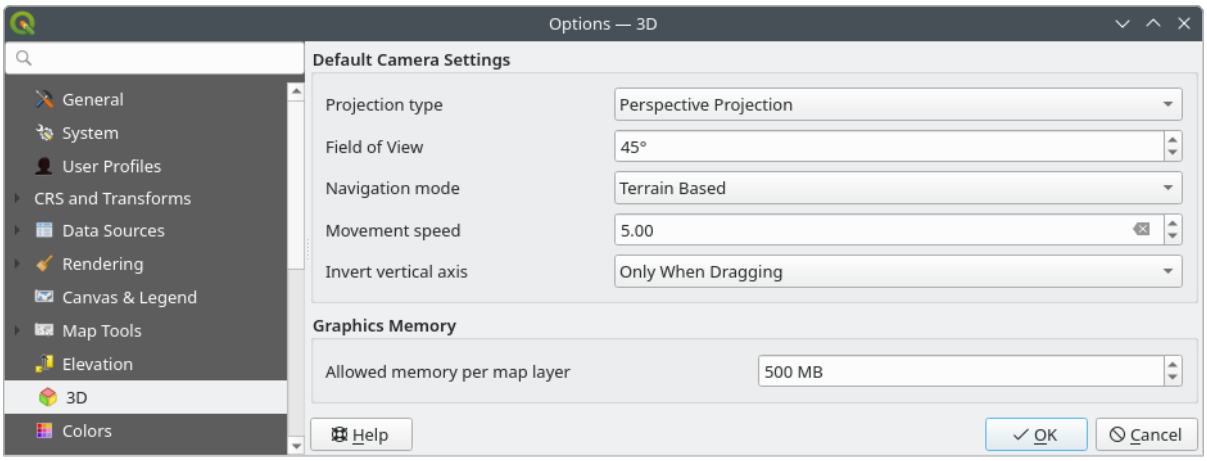
5.1.9 Aukščio nustatymai



5.16 Fig. : Aukščio nustatymai

Meniu *Aukštis* leidžia jums nurodyti spalvą, kurią reikia naudoti kaip *Fono spalvą aukščio profiliams*. Taip galite diagramas padaryti lengviau skaitomas kai kuriems duomenų rinkiniams, pavyzdžiui taškų debesims su RGB spalvinimu, kur numatytoji fono spalva yra per daug panaši į taškų spalvas, kad būtų lengvai atskiriama. Palikus nepakeistą aukščio profiliai ir toliau naudos standartinę sistemos fono spalvą.

5.1.10 3D Nustatymai



5.17 Fig. : 3D Nustatymai

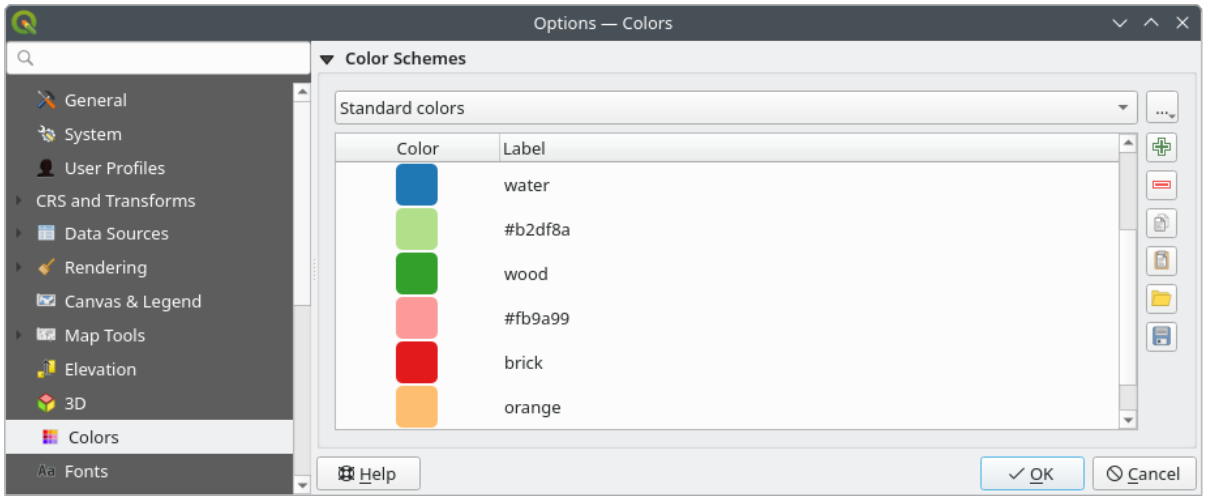
3D meniu leidžia jums konfigūruoti kai kuriuos numatytuosius nustatymus, kurie naudojami visiems *3D žemėlapio vaizdams*. Tie nustatymai gali būti:

- *Numatytieji kameros nustatymai*:
  - *Projekcijos tipas*: leidžia žiūrėti 3D sceną:
    - *Perspektyvinėje projekcijoje* (numatytoji reikšmė): Lygiagrečios linijos susikerta tolumoje. Objektai yra tuo mažesni, kuo toliau jie yra nuo kameros.
    - arba *Ortogonalioje projekcijoje*: Lygiagrečios linijos atrodo lygiagrečios. Objektai yra to paties dydžio nepriklausomai nuo to, koks jų atstumas nuo kameros.

  - Kameros *Matymo laukas*: aktualus tik perspektyvinei projekcijai, nurodo dabartinį matymo lauką laipsniais ir nurodo, kuri scenos dalis yra matoma kamerai. Numatytoji reikšmė yra 45°.
  - *Navigacijos režimas*: suteikia skirtingus būdus sąveikauti su 3D scena. Galimi režimai yra:
    * *Remiantis paviršiumi*: naviguojant kamera seka fiksuotą poziciją ant žemės paviršiaus.
    * *Vaikščiojimo režimas (pirmo asmens)*

    navigacijos komandos skirsis priklausomai nuo parinkto režimo.
  - *Judėjimo greitis*
  - *Apkeisti vertikalias ašis*: Nurodo ar vertikalios ašies judesiai turi būti sukeisti vietomis lyginant su normalia elgsena. Įtakoja tik judėjimą *Vaikščiojimo režimu*. Galima nustatyti:
    * *Niekada*
    * *Tik tempiant*: vertikalus judėjimas bus apsuktas tik darant paspaudimo ir tempimo kameros posūkį
    * ir *Visada*: judesiai bus apsukami tiek paspaudus ir tempiant, tiek ir spaudant žymeklio mygtukus (paspaudus mygtuką ~)
- Skiltyje *Grafikos atmintis*, parinktis *Vienam žemėlapio sluoksniui leidžiama atmintis* leidžia nurodyti, kiek kiekvienas sluoksnis gali naudoti GPU atminties. Tai naudinga naudotojams, kurie naudoja dideles 3D scenas, kurios išsekina turimus GPU atminties resursus. Pasiekus ribą rodomas įspėjimas, kuris padeda diagnozuoti dideles scenas.

5.1.11 Spalvų nustatymai



5.18 Fig. : Spalvų nustatymai

Šis meniu leidžia kurti ar keisti spalvų paletes, kurios naudojamos aplikacijos *spalvų parinkimo valdikliuose*. Galite pasirinkti iš:

- *Paskutinės spalvos* rodo paskutines naudotas spalvas
- *Standartinės spalvos*, numatytoji spalvų paletė
- *Projekto spalvos*, dabartiniame projekte naudojamos spalvos (daugiau informacijos rasite skyriuje *Stilių savybės*)
- *Naujo sluoksnio spalvos*, rinkinys spalvų, kurios pagal nutylėjimą naudojamos pridedant naujus sluoksnius į QGIS
- arba savo paletė(s), kurias galite kurti ar importuoti naudodami mygtuką … greta iškrentančio palečių sąrašo.

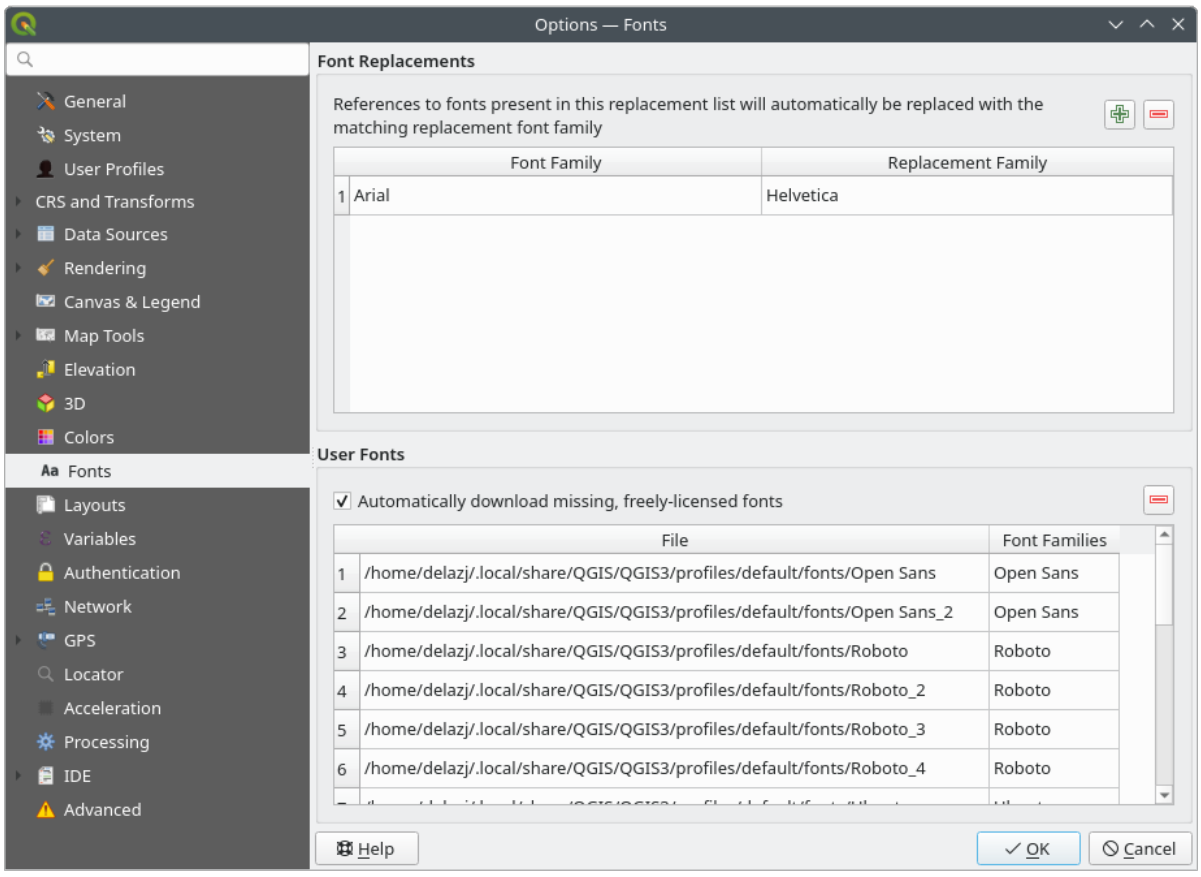
Pagal nutylėjimą *Paskutinių spalvų*, *Standartinių spalvų* ir *Projekto spalvų* palečių negalima išimti ir jos visada matosi iškrentančiame spalvų sąraše. Į šį valdiklį taipogi galima pridėti savas paletes naudojant parinktį *Rodyti spalvų mygtukuose*.

Bet kuriai iš šių palečių jūs galite tvarkyti spalvų sąrašą, naudodami greta sąrašo esančius įrankius, t.y.:

- *Pridėti* ar *Išimti* spalvą
- *Kopijuoti* ar *Įkelti* spalvą
- *Importuoti* ar *Eksportuoti* spalvų rinkinį iš/į `.gpl` failą.

Du kartus paspaudus ant spalvos sąraše, galite ją tvarkyti ar pakeisti *Spalvų parinkimo* dialoge. Taip pat galite ją pervadinti du kartus paspaudę ant *Etiketės* stulpelio.

5.1.12 Šriftų nustatymai



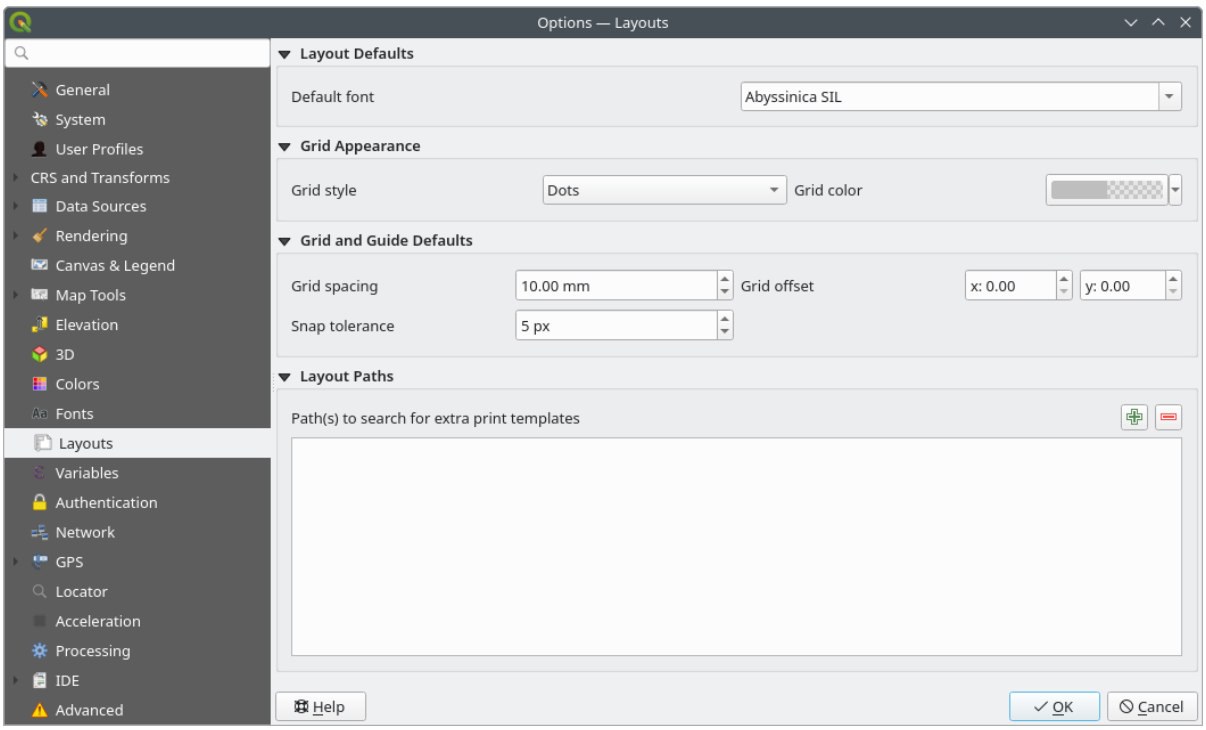
5.19 Fig. : Šriftų nustatymai

Kortelė *Šriftai* leidžia valdyti šriftus projektuose:

- *Šriftų pakeitimai*: leidžia nurodyti automatiškus šriftų keitimus, naudojamus įkeliant projektus ar stilius, taip pagerinant palaikymą skirtingose operacinėse sistemose (pvz. keisti „Arial“ į „Helvetica“).
- *Naudotojo šriftai*: Leidžia jums padėti TTF ar OTF šriftus į *naudotojo profilio* poaplankį `fonts`. Šie šriftai bus automatiškai įkelti QGIS paleidimo metu. Tai leidžia naudoti šriftus, kurie nėra įdiegti operacinės sistemos lygmenyje, kas dažniausiai būna blokuojama įmonių aplinkose. Skydelyje išvardinti visi įdiegti šriftai ir leidžiama tvarkyti (pvz. išimti) anksčiau įdiegtus naudotojo šriftus.

Taipogi galima *Automatiškai atsiųsti trūkstamus laisvai licencijuotus šriftus*: Pvz. jei atidarote projektą ar stilių, ar bandote įkelti vektorinių kaladėlių sluoksnį, kurie mini šriftus, kurių šiuo metu sistemoje nėra, tada pagal įdėtą laisvai prieinamų šriftų sąrašą patikrinama, ar galima reikiamą šriftą atsisiųsti į naudotojo profilio šriftų aplanką (su šrifto licencijos pranešimu).

5.1.13 Išdėstymų nustatymai



5.20 Fig. : Išdėstymų nustatymai

Maketo numatytieji

Galite nurodyti *Numatytąjį teksto šriftą*, kuris bus naudojamas *spausdinimo išdėstyme*.

Tinklelio išvaizda

- Nustatykite *Tinklelio stilių* (‚Ištisinis‘, ‚Taškai, ‚Kryžiukai‘)
- Nurodykite *Tinklelio spalvą*

Numatytieji tinklelio ir orientacinių linijų parametrai

- Nustatykite *Tinklelio tarpus*
- Nustatykite *Tinklelio poslinkį* X ir Y
- Nustatykite *Pritraukimo toleranciją*

Maketų keliai

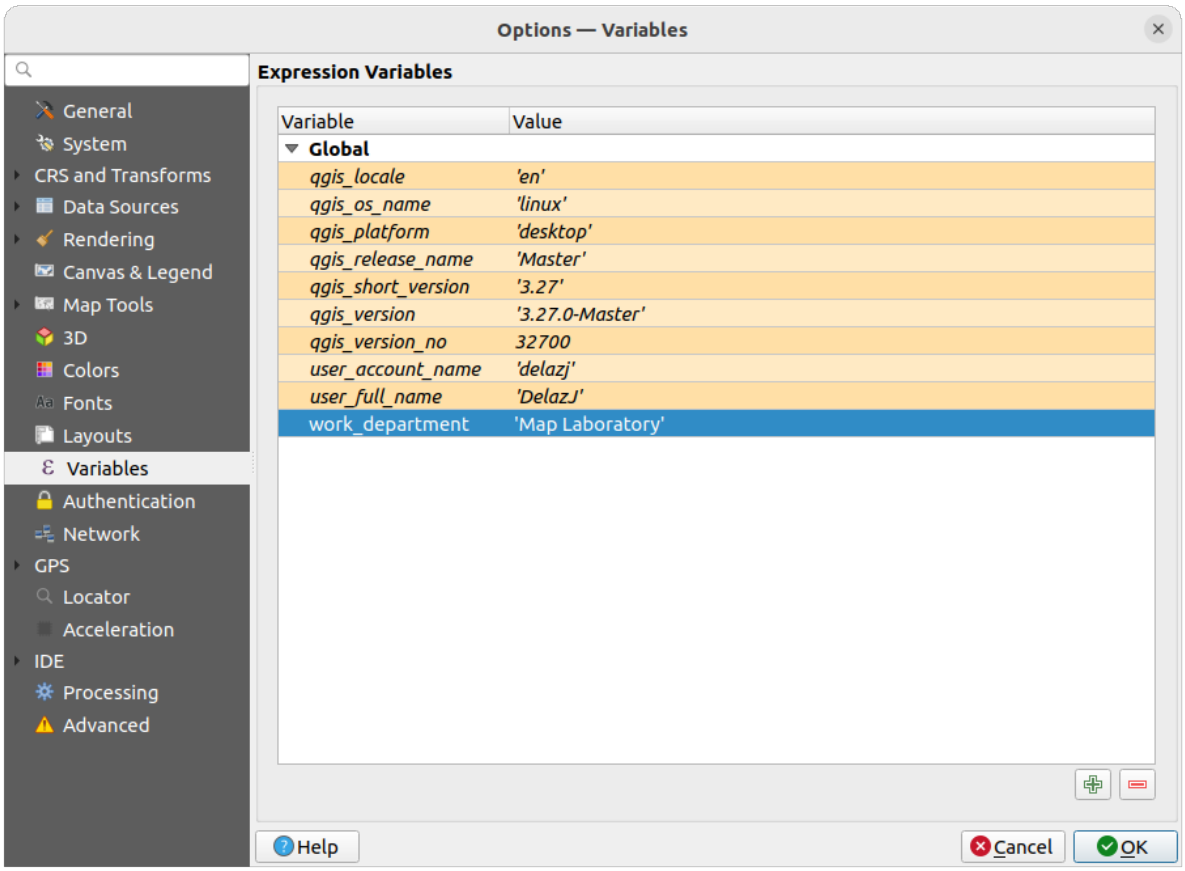
- Nurodykite *Kelią(us), kuriuose ieškoti spausdinimo šablonų*: sąrašas aplankų su savai išdėstymų šablonais, naudojamais kuriant naują išdėstymą.

5.1.14 Kintamųjų nustatymai

Kortelėje *Kintamieji* yra globalaus lygio kintamųjų sąrašas.

Čia taip pat galima valdyti globalaus lygio kintamuosius. Norėdami pridėti naują savo globalų kintamąjį, spauskite mygtuką . Analogiškai parinkite sąraše savo globalų kintamąjį ir spauskite , kad jį išimtumėte.

Daugiau informacijos apie kintamuosius rasite skyriuje *Reikšmių saugojimas kintamuosiuose*.



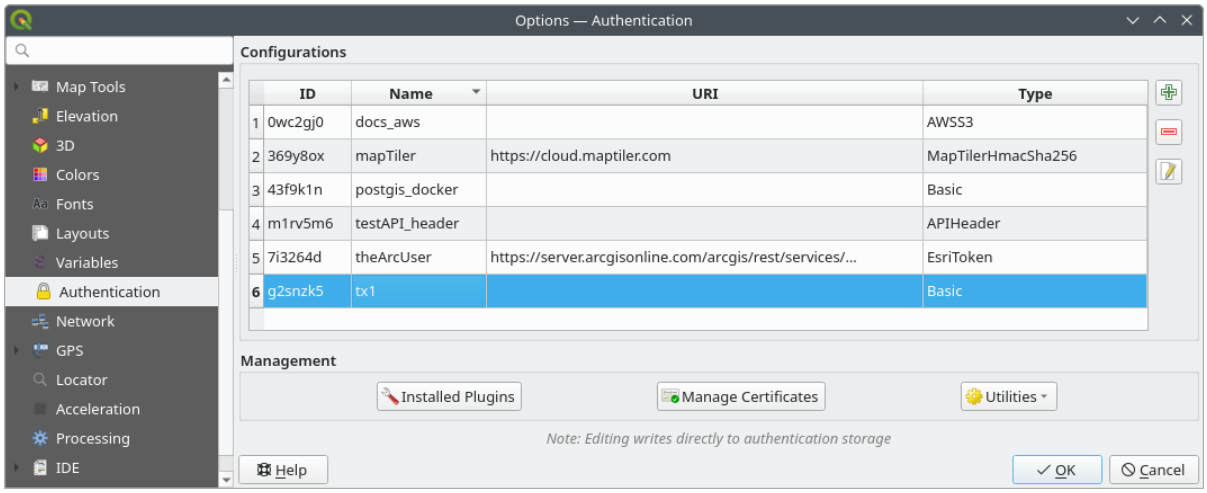
5.21 Fig. : Kintamųjų nustatymai

5.1.15 Autentifikacijos nustatymai

Kortelėje *Autentifikacija* jūs galite kurti autentifikacijos konfigūracijas ir valdyti PKI sertifikatus. Daugiau informacijos rasite skyriuje *Autentifikacijos sistema*.

Autentifikacijų tvarkymui jūs galite naudoti šalia lentelės esančių įrankių sąrašą:

- Pridėti naują autentikacijos konfigūraciją
- Išimti pažymėtą autentikacijos konfigūraciją
- Keisti pažymėtą autentikacijos konfigūraciją

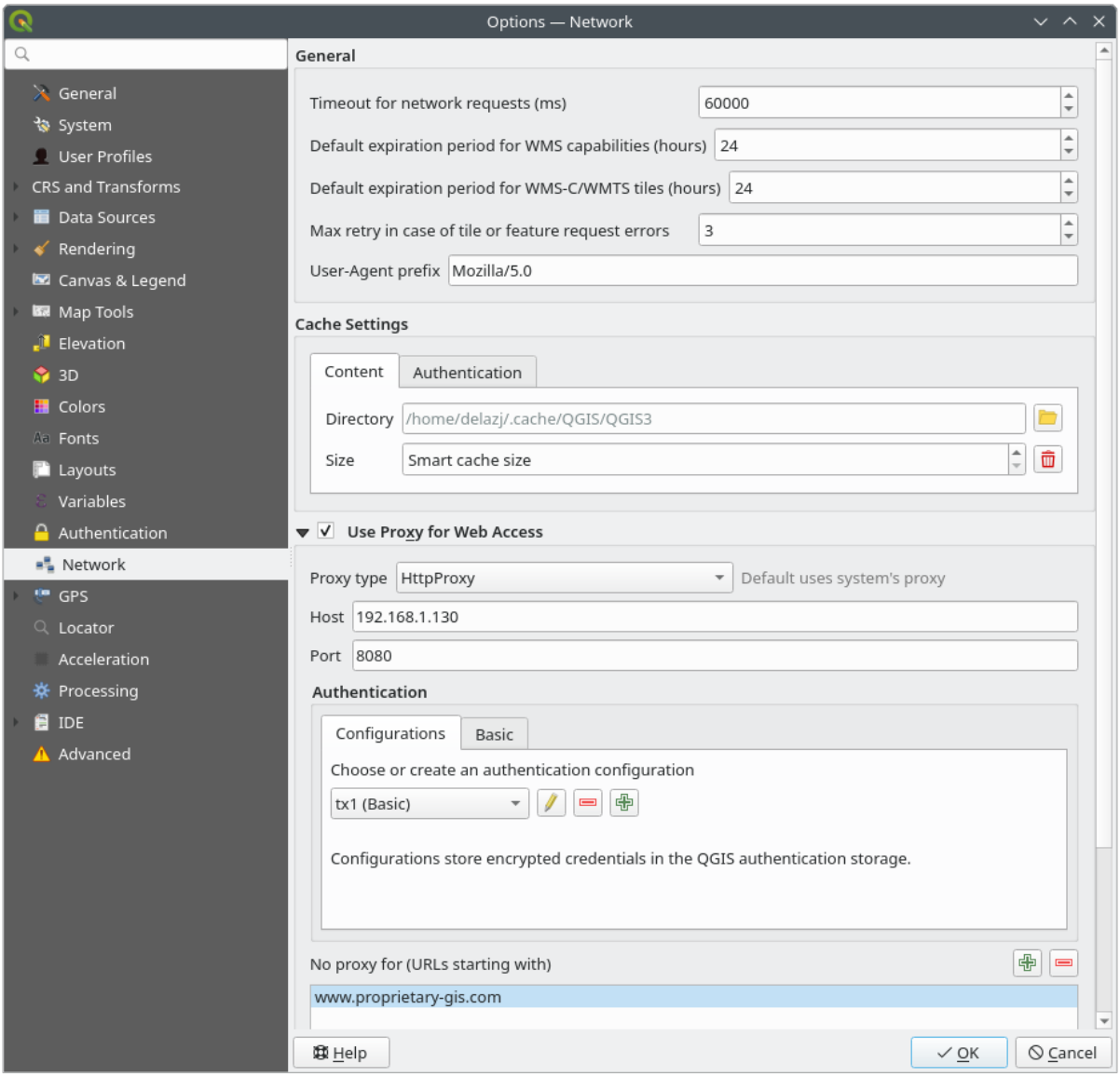


5.22 Fig. : Autentifikacijos nustatymai

5.1.16 Tinklo nustatymai

Bendra

- Nustatykite kiek *Tinklo užklausų atsakymo laukti (ms)* - numatytoji reikšmė yra 60000
- Nustatykite *Numatytą pasenimo periodą WMS galimybėms (valandomis)* - numatytoji reikšmė yra 24
- Nustatykite *Numatytą pasenimo periodą WMS-C/WMTS kaladėlėms (valandomis)* - numatytoji reikšmė yra 24
- Nustatykite, kiek *Daugiausia pakartojimų esant kaladėlės ar geoobjekto užklausos klaidai*
- Nustatykite *Naudotojo-agento priešdėlį*, kurio numatytoji reikšmė yra `Mozilla/5.0`. Ši reikšmė bus pridedama prie QGIS ir OS versijos, pvz., `Mozilla/5.0 QGIS/33801/Ubuntu 22.04.4 LTS`, kad būtų apibrėžtas naudotojo agentas.



5.23 Fig. : Tinklo ir šliuzo nustatymai

Podėlio nuostatos

Apibrėžia podėlio *Aplanką* ir *Dydį* megabaitais. Jūs galite išvalyti reikšmę, kad įjungtumėte *Išmanų podėlio dydį*, kuris nustato maksimalų podėlio dydį pagal laisvą vietą diske. Taipogi siūlo įrankius, skirtus *automatiškai išvalyti jungties autentifikacijos podėlį įvykus SSL klaidoms (rekomenduojama)*.

Interneto prieigai naudoti šliuzą

- *Interneto prieigai naudoti šliuzą*
- Nustatykite *Šliuzo tipą* pagal savo reikalavimus ir nustatykite „Stotį“ ir „Prievadą“. Galimi šliuzo tipai yra:
  - *Numatytasis šliuzas*: Šliuzas nustatomas pagal sistemos šliuzą
  - *Socks5Proxy*: Bendrinis šliuzas bet kokiai jungčiai. Palaiko TCP, UDP, pririšimą prie prievado (įeinančioms jungtims) ir autentifikaciją.
  - *HttpProxy*: Įgyvendinama naudojant komandą „CONNECT“, palaiko tik išeinančias TCP jungtis; palaiko autentifikaciją.

- *HttpCachingProxy*: Įgyvendinta naudojant paprastas HTTP komandas, naudinga tik HTTP užklausų kontekste.
- *FtpCachingProxy*: Įgyvendinta naudojant FTP šliužą, naudinga tik FTP užklausų kontekste.

Šliuzo prisijungimo duomenys nustatomi *autentifikacijos valdiklyje*.

Žemiau esančiame tekstiniame šliuzo nustatymų lauke galima pridėti kelis išimčių URL (žr. Fig. 5.23). Jei paskirties url prasideda viena iš nurodytų eilučių, nebus naudojamas joks šliuzas.

Jei jums reikia detalesnės informacijos apie skirtingų šliuzų nustatymus, prašome skaityti atitinkamą QT bibliotekos dokumentaciją https://doc.qt.io/archives/qt-5.9/qnetworkproxy.html#ProxyType-enum

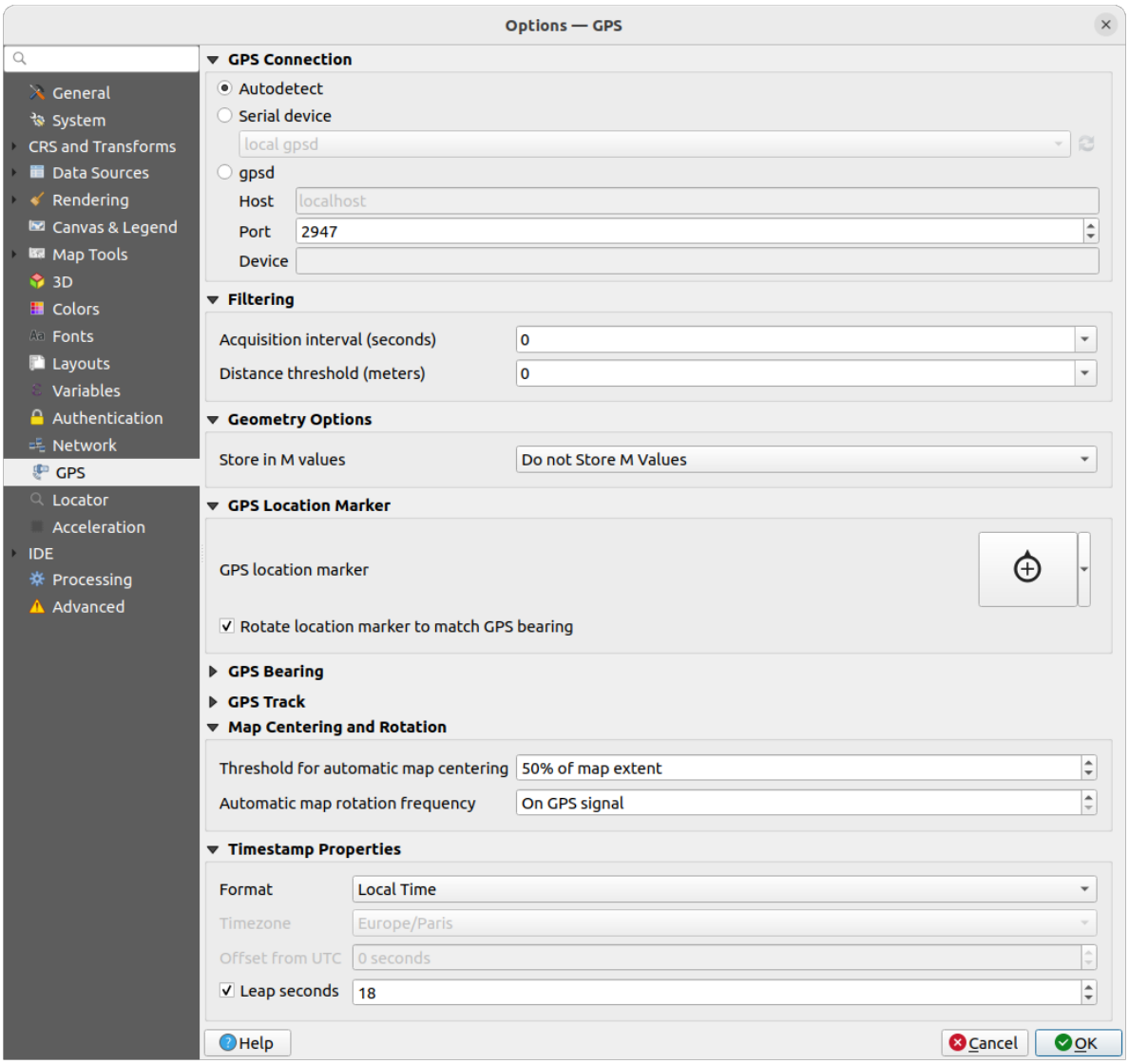
Patarimas

Šliuzo naudojimas

Šliuzo naudojimas kartais gali būti sudėtingas. Naudinga veikti „bandymų ir klaidų“ metodu su aukščiau išvardintais šliuzų tipais, tikrinant kurie tinka jūsų atvejui.

5.1.17 GPS nustatymai

GPS dialogas padeda jums konfigūruoti GPS įrenginių jungtis ir QGIS savybes. Jis taip pat teikia GPS sekimo ir duomenų skaitmeninimo nustatymus.



5.24 Fig. : GPS nustatymai

Jūs galite nurodyti:

- *GPS jungtis*: teikia įvairius būdus prijungti įrenginį prie QGIS
  - *Autoaptikti*
  - *Serijinis įrenginys* (prijungus naują GPS įrenginį reikės iš naujo įkelti)
  - *gpsd* (parinkite *Stotį* ir *Prievadą*, prie kurio prijungtas jūsų GPS, bei nurodykite *Įrenginio* pavadinimą)

  Jei bus jungties problemų, galite bandyti persijungti nuo vieno prie kito.
- *Filtravimas*: Jūs galite nurodyti *Gavimo intervalo (sekundėmis)* ir/arba *Atstumo slenksčio (metrais)* parametrus, kad žymeklis liktų aktyvus, kai gaviklis yra statinėje padėtyje.
- *Geometrijos parinktyse*, parinktis *Įrašyti M reikšmes* leidžia kurti geometrijas su M reikšmėmis iš vidinių GPS žurnalizavimo įrankių. Tai taikoma tiek geoobjektams, suskaitmenintiems iš GPS žurnalo, tiek ir iš *Žurnalizavimo Geopackage/SpatiaLite* funkcionalumo… Parinktyse yra galimybė įrašyti laiko žymų (kaip ms nuo epochos), žemės greičio, aukščio, krypties ir tikslumo komponentus kaip m reikšmes.
- *GPS vietos žyma*

  - Konfigūruoja *GPS vietos žymeklio* simbolį, kuris rodo dabartinę GPS poziciją
  - *Pasukti vietos žymeklį, kad atitiktų GPS kryptį*: ar reikia pasukti žymeklio simbolį taip, kad jis atitiktų iš GPS gaunamą kryptį
- *GPS kryptis*:
  - konfigūruokite *Krypties linijos stilių*, naudojant QGIS linijų simbolių savybes
  - įjunkite *Skaičiuoti kryptis pagal kelionės kryptį*: Įjungus šią parinktį, kryptis, gauta iš GPS įrenginio, bus ignoruojama, o vietoje to kryptis bus išskaičiuojama naudojant dvi paskutines GPS vietas.
- *GPS pėdsakas*: nustatykite simbolį, kuris turėtų būti naudojamas *Pėdsako linijos stiliui*
- *Žemėlapio centravimas ir pasukimas*: nurodo, kada žemėlapio drobė atnaujinama pagal naudotojo postūmį vietovėje:
  - *Slenkstis automatiniam žemėlapio centravimui*: nurodo minimalų GPS pozicijos poslinkį nuo žemėlapio drobės centro, kad būtų sukeltas automatinis *žemėlapio perpaišymas*, jei įjungtas.
  - *Automatinio žemėlapio pasukimo dažnis*: nurodo kaip dažnai pasukti žemėlapį, kad jis atitiktų GPS kryptį; tai gali būti savo laiko intervalas arba:*Pagal GPS signalą*.
- *Laiko žymos savybės* leidžia konfigūruoti, kaip rodomos ir saugomos duomenyse laiko reikšmės. Tarp parametrų yra ir *Formatas*, kuris gali būti:
  - *Vietinis laikas*
  - *UTC*
  - *UTC su poslinkiu*, kad būtų atsižvelgta į laiko sukiojimą ar kitus sudėtingus laiko juostos dalykus
  - konkreti *Laiko juosta*

  Papildomai galima taikyti *Keliamųjų sekundžių* korekciją, pridedant sekundes prie GPS laiko žymos.

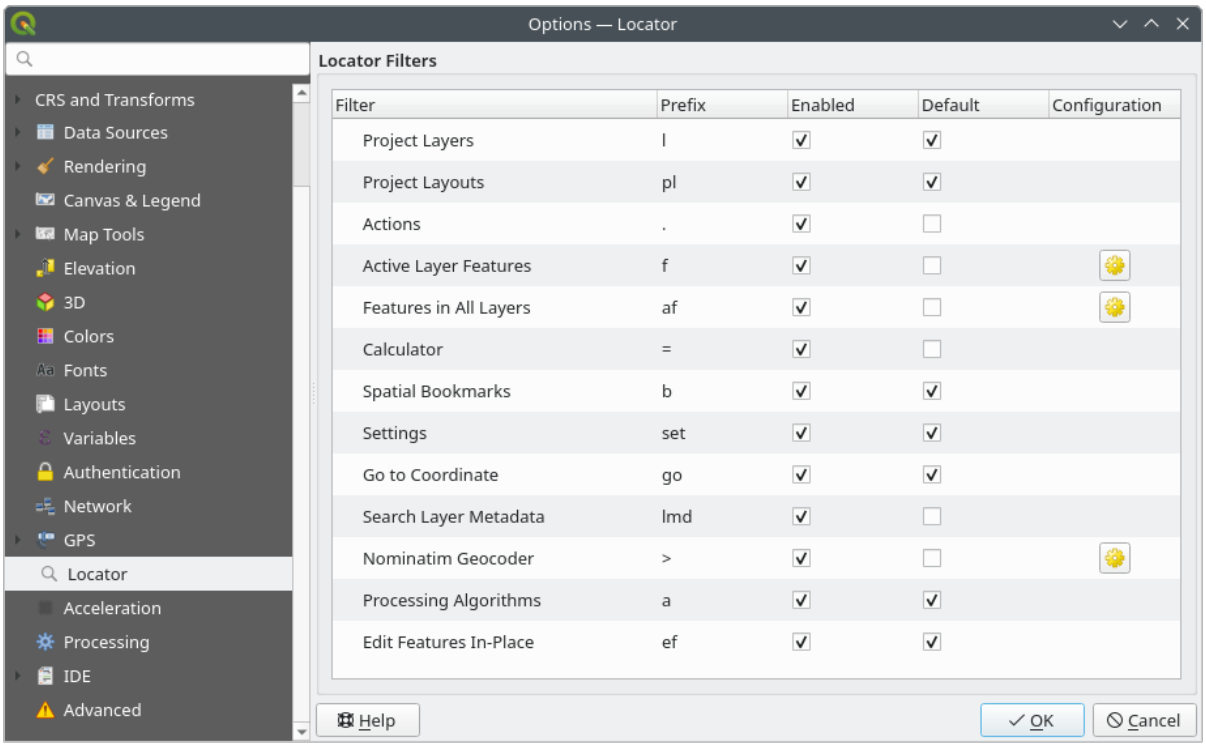
GPSBabel

GPSBabel keičia kelio taškus, pėdsakus ir maršrutus tarp populiarių GPS imtuvų, tokių kaip Garmin ar Magellan bei žymėjimo programų, tokių kaip Google Earth ar Basecamp formatų. Palaikomi šimtai GPS imtuvų bei programų. QGIS priklauso nuo GPSBabel bendravimui su šiais įrenginiais ir *jų duomenų manipuliavimui*.

Daugiau informacijos rasite *Loading to or from a device*.

5.1.18 Atradėjo nustatymai

Kortelė *Atradėjas* leidžia konfigūruoti *Atradėjo juostą*, greitos paieškos valdiklis būsenos juostoje, kuris leidžia vykdyti paiešką aplikacijoje. Jis teikia kai kuriuos numatytuosius filtrus (su priešdėliu):



5.25 Fig. : Atradėjo nustatymai

- *Projekto sluoksniai* (`l`): randa ir pažymi sluoksnius *Sluoksnių* skydelyje.
- *Projekto išdėstymai* (`pl`): randa ir atidaro spausdinimo maketą.
- Veiksmai (`.`): randa ir vykdo QGIS veiksmą; veiksmu gali būti bet kuris įrankis ar QGIS meniu, skydelio atidarymas…
- *Aktyvaus sluoksnio geoobjektai* (`f`): ieško atitinkančių atributų bet kuriame aktyvaus sluoksnio lauke ir priartina pasirinktą geoobjektą. Spauskite , kad sukonfigūruotumėte maksimalų rezultatų skaičių.
- *Geoobjektai visuose sluoksniuose* (`af`): ieško atitinkančių atributų *ieškomų sluoksnių rodomame pavadinime* ir priartina pažymėtą geoobjektą. Spauskite , kad sukonfigūruotumėte bendrą maksimalų rezultatų skaičių ir rezultatų skaičių vienam sluoksniui.
- *Skaičiuotuvas* (`=`): leidžia įvertinti bet kokią QGIS išraišką ir leidžia nukopijuoti rezultatą į iškarpinę.
- *Erdvinės žymelės* (`b`): randa ir priartina žymelės apimtį.
- *Nustatymai* (`set`): naršo ir atidaro projekto ir aplikacijos savybių dialogus.
- *Eiti į koordinatę* (`go`): pastumia žemėlapio drobę į vietą, apibrėžta kableliais ar tarpu atskirtomis x ir y koordinatėmis arba suformuotu URL (pvz., OpenStreetMap, Leaflet, OpenLayer, Google Maps, …). Koordinačių tikimasi WGS 84 (`epsg:4326`) ir/arba žemėlapio drobės CRS.
- *Nominatim Geocoder* (`>`): geocodes using the Nominatim geocoding service of the OpenStreetMap Foundation. Press to limit results to one or more countries using list of country codes. This helps narrow down the search to specific countries, avoiding ambiguous results from other locations. For example, entering `DE` or `DE,FR` restricts results to Germany or Germany and France, respectively.
- Apdorojimo algoritmai (`a`): ieško ir atidaro Apdorojimo algoritmo dialogą.
- *Keisti parinktus geoobjektus* (`ef`): leidžia greitą prieigą ir paleidžia suderinamą *redaguoti vietoje* Apdorojimo algoritmą aktyviam sluoksniui.

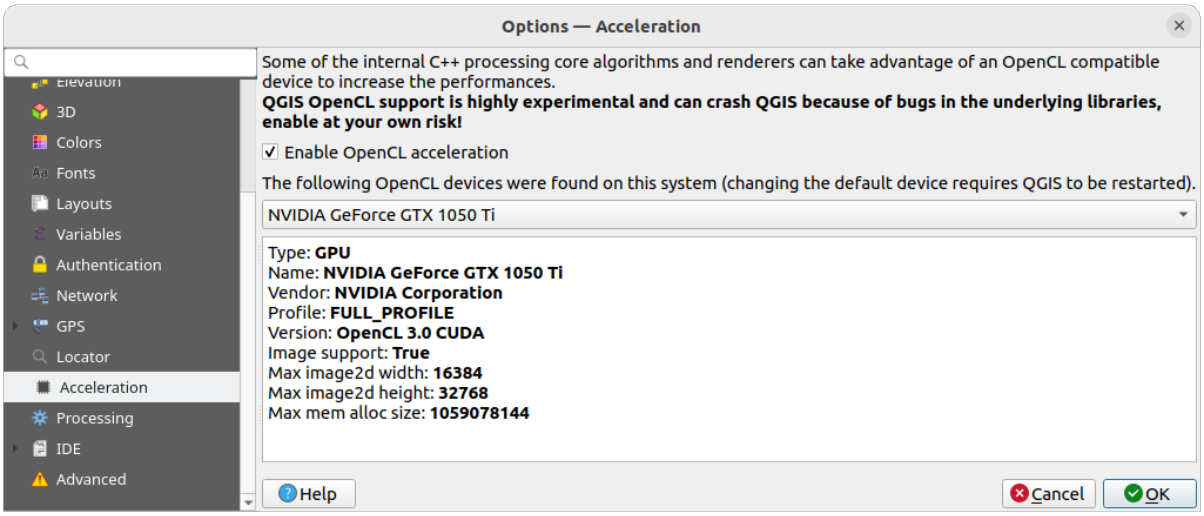
Dialoge jūs galite:

- keisti filtro *Priešdėlį*, t.y. raktinį žodį filtro iškvietimui
- nustatyti, ar filtras yra *Įjungtas*: filtras gali būti naudojamas paieškoms ir jo klavišų kombinacija gali būti prieinama atradėjo juostos meniu
- nustatyti,ar filtras yra *Numatytasis*: paieška be filtro grąžina rezultatus tik iš numatytųjų filtro kategorijų.
- Kai kurie filtrai suteikia būdą konfigūruoti paieškos rezultatų skaičių.

Numatytųjų atradėjo filtrų rinkinys gali būti išplėstas priedais, pavyzdžiui OSM nominatim paieškai, tiesioginei duombazės paieškai, sluoksnių katalogų paieškoms, …

5.1.19 Greitinimo nustatymai

OpenCL greitinimo nustatymai.



5.26 Fig. : Greitinimo nustatymai

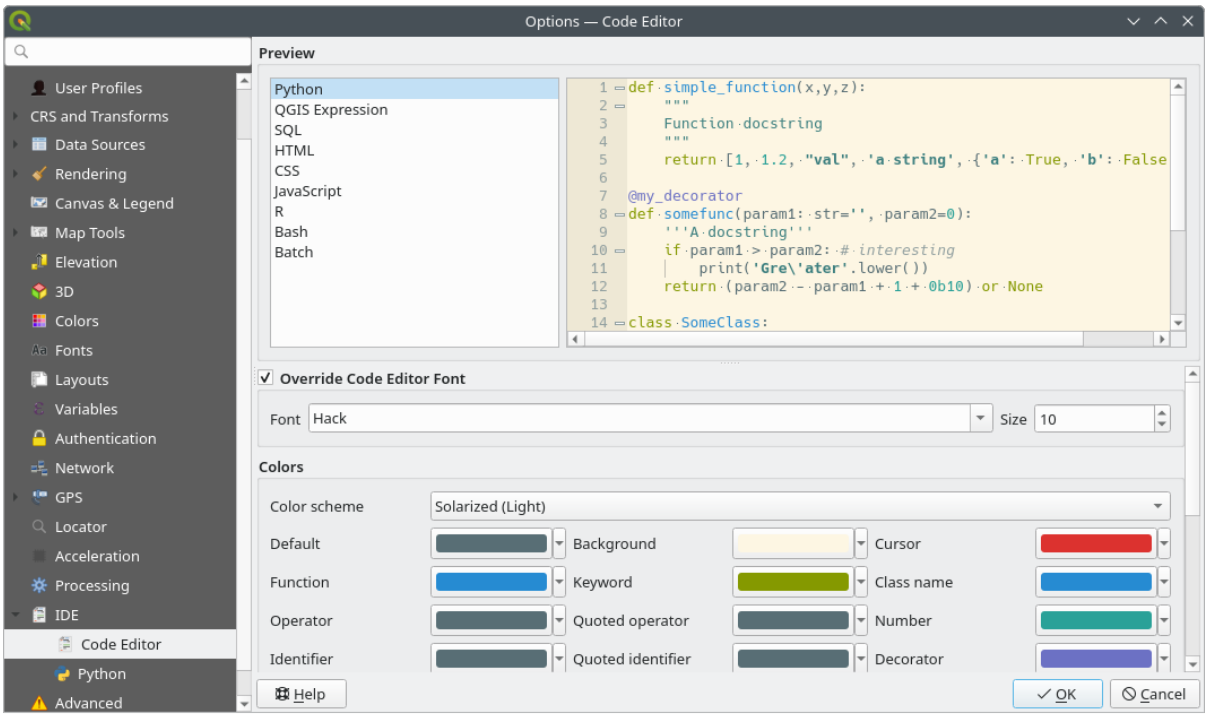
Priklausomai nuo jūsų techninės ir programinės įrangos, jums gali reikėti įdiegti papildomas bibliotekas, kad įgalintumėte OpenCL greitinimą.

5.1.20 IDE nustatymai

GitHub prieigos žetonas, jūs galite sukurti asmeninį žetoną, leidžiantį jums dalintis kodo gabaliukais iš Python kodo redaktoriaus. Daugiau informacijos rasite dokumente GitHub autentifikacija

Kodo redaktoriaus nustatymai

Kodo redaktoriaus kortelėje jūs galite valdyti redagavimo valdiklių (Pythono interaktyvios konsolės ir redaktoriaus, išraiškų valdiklio bei funkcijų redaktoriaus) išvaizdą ir elgseną.



5.27 Fig. : Kodo redaktoriaus nustatymai

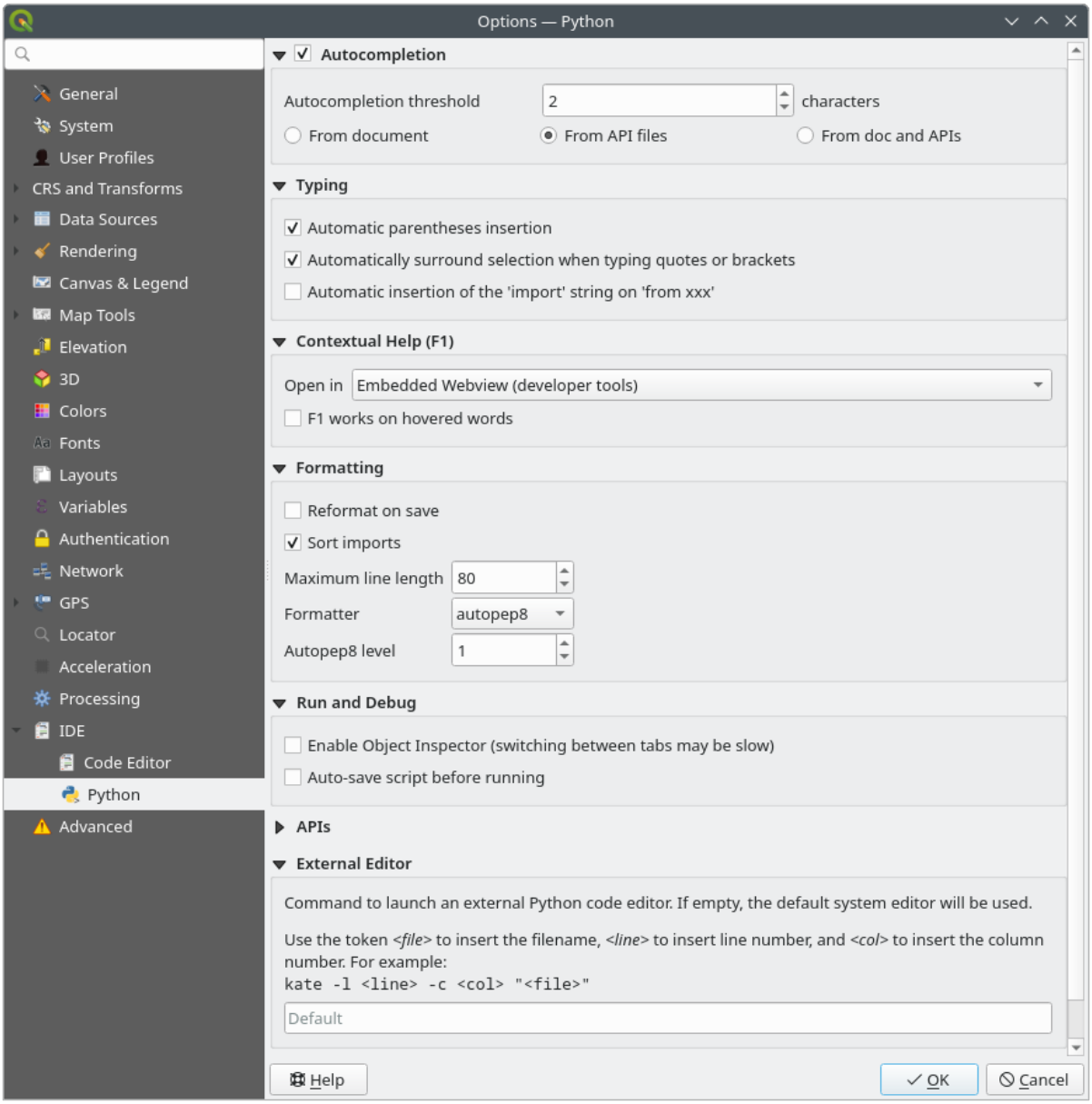
Šio dialogo viršuje valdiklis suteikia gyvą dabartinių nustatymų peržiūrą skirtingomis programavimo kalbomis (Python, QGIS išraiškos, HTML, SQL, JavaScript). Patogus būdas keisti nustatymus.

- Įjunkite parinktį *Permušti kodo redaktoriaus šriftą*, jei norite pakeisti numatytąją *Šrifto* šeimą ir *Dydį*.
- Grupėje *Spalvos* jūs galite:
  - parinkti *Spalvų schemą*: numatytieji nustatymai yra `Numatytoji`, `Soliarizuota tamsi` ir `Soliarizuota šviesi`. `Sava` schema įjungiama iš karto, kai jūs pakeičiate spalvą ir ją visada galima atšaukti parinkus iš anksto numatytą schemą.
  - keisti kiekvieno kodo rašymo elemento *spalvą*. Pavyzdžiui galite pakeisti komentarų, kabučių, funkcijų ar fono spalvą…

Pythono nustatymai

Python nustatymai padeda tvarkyti ir valdyti Python redaktorių elgseną (*inetraktyvi konsolė*, *kodo redaktorius*, *projekto makro*, *savos išraiškos*, …). Ji taipogi pasiekiama naudojant mygtuką Parinktys… iš:

- *Python konsolės* įrankinės
- *Python konsolės* valdiklio kontekstinio meniu
- ir kodo redaktoriaus kontekstinio meniu.



5.28 Fig. : Pythono nustatymai

Jūs galite nurodyti:

- *Autocompletion*: Enables code completion. You can get autocompletion from the current document, the *installed API* files or both.
  - *Autocompletion threshold*: Sets the threshold for displaying the autocompletion list (in number of characters)
- *Rašoma*
  - *Automatinis skliaustelių įterpimas*: Jei nepažymėtas joks tekstas ir įvedamas atidarymo simbolis (skliausteliai, kabutės, laužtiniai skliaustai, …), iš karto už žymeklio įterpiamas uždarantis simbolis. Pastebėtina, kad ši elgsena išjungta, kai dabartinis žymeklis yra eilutės ar komentaro viduje.
  - *Automatiškai apgaubti pažymėtą vietą rašant kabutes ar skliaustus*: Kai įvedamas atidarantis simbolis, pažymėtas tekstas yra apgaubiamas atidarymo/uždarymo pora. Pažymėjimas išlieka, taigi galima apgaubti kabutėmis pažymėtą žodį ir tada apgaubti jį skliausteliais tiesiog spaudžiant `"`, o tada `(`.

Specialus atvejis yra kelių eilučių pažymėjimas su kabutėmis ir dvigubomis kabutėmis: pažymėjimas apgaubiamas trigubomis viengubomis/dvigubomis kabutėmis.

  - ☐ *Automatinis „import“ eilutės įterpimas parašius „from xxx“*: Įgalina „import“ įterpimą nurodant importus

- *Contextual Help (F1)* provides options to control QGIS behavior when requesting help on a PyQGIS or Qt object in a Python code sample. When pressing `F1`, QGIS will open the API page (or a search results page) matching:

  1. the selected text if any OR
  2. the word at the mouse position (if ☑ *F1 works on hovered words* is enabled) OR
  3. the word at the text cursor position

  This works for both Qt and PyQGIS API documentation. The page is opened either in QGIS, as an *Embedded webview (developer tools)*, or in the *Default system web browser*.

- Skiltyje *Formatavimas* jūs galite pridėti automatinius įrankius jūsų kodo performatavimui:
  - ☐ *Performuoti įrašant*: formatavimas taikomas prieš pat įrašant jūsų scenarijų
  - ☑ *Rikiuoti importus*: rikiuoja ‚import‘ sakinius naudojant isort biblioteką
  - *Maksimalus eilutės ilgis*: valdo formatuotojo ilgų eilučių perkėlimą bei redaktoriaus liniuotę
  - *Formatuotojas* - palaikomi įrankiai yra *autopep8* ir *black*, su tam skirta parinktimi:
    * *Autopep8 lygis* - daugiau informacijos apie autopep8 išmanesnį naudojimą
    * ☐ *Normalizuoti kabutes*: keičia visas viengubas kabutes dvigubomis kabutėmis, jei tik galima
- *Vykdymas ir tinkinimas*
  - ☐ *Įjungti objektų inspektorių (persijungimas tarp kortelių gali būti lėtas)*
  - ☐ *Automatiškai įrašyti prieš paleidžiant*: Vykdant automatiškai įrašo skriptą. Šis veiksmas įrašys laikiną failą (laikiname sistemos aplanke), kuris bus automatiškai ištrintas po įvykdymo.

Under *APIs*, you can specify which APIs your code should rely on:

- ☑ *Using preloaded APIs file*: You can choose if you would like to use the preloaded API files. By default, supported APIs are:
  - Python
  - PyQGIS
  - PyQt
  - QScintilla2
  - osgeo-gdal-ogr
  - osgeo-geos

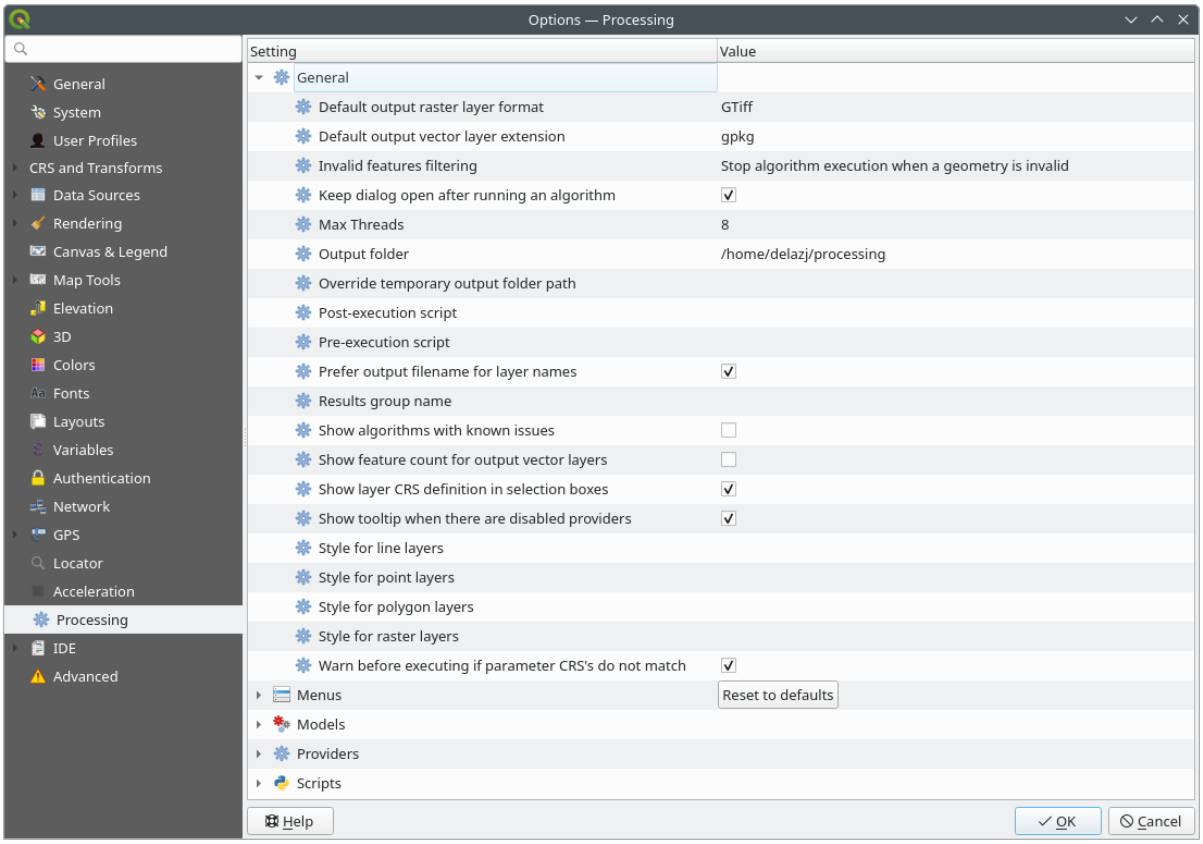
  You can also choose to add specific API files.

- ☐ *Naudojamas paruoštas API failas*: Įjungus šią parinktį, parinktas `*.pap` failas bus naudojamas kodo kompiliavimui. Norėdami sukurti paruoštą API failą turite įkelti bent vieną `*.api` failą ir tada sukompiliuoti jį paspausdami mygtuką *Kompiliuoti API…*.

Išorinio redaktoriaus grupė leidžia jums teikti komandinės eilutės instrukcijas, kurių pagalba paleidžiamas išorinis Python kodo redaktorius, pagal nurodytą failo pavadinimą, eilutės ir stulpelio numerį. Palikus tuščią, naudojamas numatytasis sistemos redaktorius.

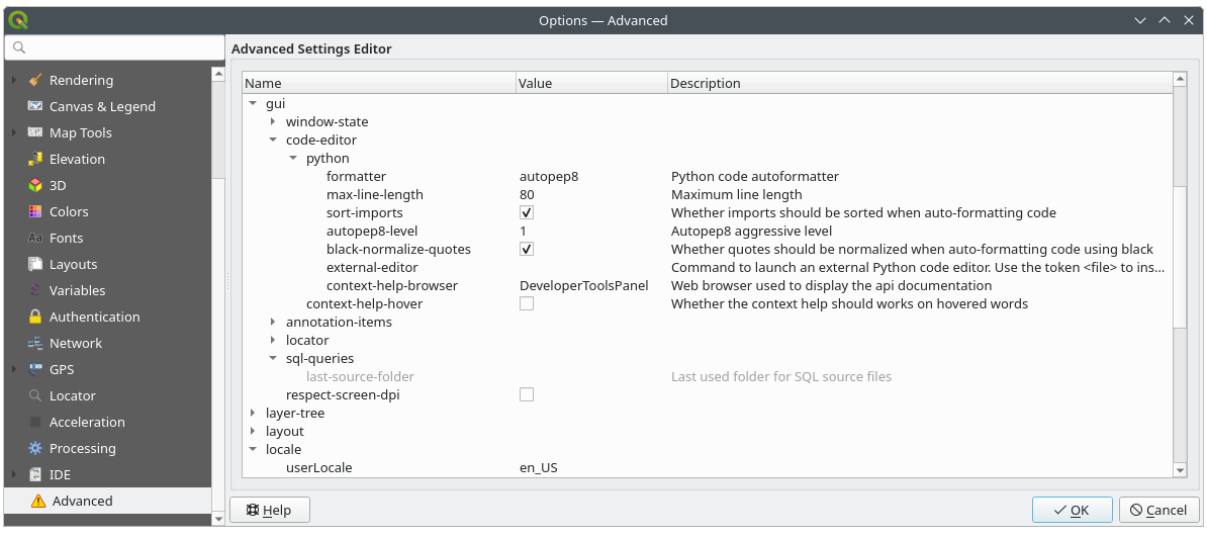
5.1.21 Apdorojimo nustatymai

Kortelė *Apdorojimas* teikia pagrindinius įrankių ir duomenų teikėjų, naudojamų QGIS apdorojimo karkase, nustatymus. Daugiau informacijos rasite skyriuje *Apdorojimo karkaso konfigūravimas*.



5.29 Fig. : Apdorojimo nustatymai

5.1.22 Sudėtingesni nustatymai



5.30 Fig. : Sudėtingesni nustatymai

Visi šie nustatymai susiję su QGIS (naudotojo sąsaja, įrankiai, duomenų tiekėjai, apdorojimo konfigūracija, numatytosios reikšmės ir keliai, priedų parinktys, išraiškos, geometrijų tikrinimai…), jie įrašomi į `QGIS/QGIS3.ini` failą aktyvaus *naudotojo profilio* aplanke. Konfigūracijomis galima dalintis nukopijuojant jas į kitų įdiegtų aplikacijų aplankus.

Iš QGIS, kortelė *Sudėtingesni* leidžia valdyti šiuos nustatymus per *Sudėtingesnių nustatymų redaktorių*. Pažadėjus būti atsargiais, valdiklyje užpildomas visų esamų nustatymų medis ir jūs galite keisti jų reikšmes. Spauskite dešinį pelės mygtuką ant nustatymo ar grupės ir jūs galėsite jį ištrinti (norėdami pridėti nustatymą ar grupę jūs turite redaguoti `QGIS3.ini` failą). Pakeitimai automatiškai įrašomi į `QGIS3.ini` failą.

Įspėjimas

Venkite aklo Sudėtingesnių nustatymų kortelės naudojimo

Būkite labai atsargūs keisdami šio dialogo elementus, nes jie pritaikomi automatiškai. Darant pakeitimus nesuprantant jų reikšmės galima įvairiais būdais sulaužyti jūsų QGIS aplikaciją.

5.2 Darbas su naudotojo profiliais

5.2.1 Koncepcija

Meniu *Nustatymai* ► *Naudotojo profiliai* leidžia nustatyti ir parinkti naudotojų profilius. Naudotojo profilis - tai bendra aplikacijos konfigūracija, viename aplanke saugant:

- visus *globalius nustatymus*, įskaitant lokalę, projekcijas, autentifikacijos nustatymus, spalvų paletes, klavišų kombinacijas…
- GUI konfigūracijas ir *tinkinimą*
- grid failus ir kitus proj pagalbinius failus, įdiegtus iš duomenų transformacijos
- įdiegtus *priedus* ir jų konfigūracijas
- projektų šablonus ir įrašytų projektų istoriją su peržiūros nuotraukomis
- *apdorojimo nustatymus*, žurnalus, scenarijus ir modelius.

Pagal nutylėjimą QGIS diegime yra vienas naudotojo profilis vardu `default`. Bet jūs galite sukurti tiek naudotojų profilių, kiek norite:

1. Spauskite *Nustatymai* ► *Naudotojo profiliai* ► *Naujas profilis…*.
2. Jūsų paprašys įvesti profilio vardą. Bus sukurtas tokio paties pavadinimo aplankas `~/<UserProfiles>/` kur:
   - `~` reprezentuoja **NAMŲ** aplanką, kuris Windows sistemoje paprastai yra kažkas panašaus į `C:\Users\<username>`.
   - o `<UserProfiles>` reprezentuoja pagrindinį profilių aplanką, pvz.:
     - `.local/share/QGIS/QGIS3/profiles/`
     - `%AppData%\Roaming\QGIS\QGIS3\profiles\`
     - `Library/Application Support/QGIS/QGIS3/profiles/`

   Naudotojo profilio aplanką galima atidaryti iš QGIS naudojant *Atverti aktyvaus profilio aplanką*.
3. Paleidžiama nauja QGIS kopija naudojant švarią konfigūraciją. Čia jūs galite nustatyti savo konfigūracijas.

Jei turite daugiau nei vieną naudotojo profilį, aktyvaus profilio pavadinimas rodomas aplikacijos antraštėje laužtiniuose skliaustuose.

Kadangi kiekviename naudotojų profilyje yra izoliuoti nustatymai, priedai ir istorija, jie gali būti puikiai panaudoti skirtingoms darbų sekoms, demonstracijoms, skirtingiems tos pačios mašinos naudotojams arba nustatymų testavimui ir pan. Persijungti iš vieno profilio į kitą galite jį parinkdami meniu *Nustatymai* ► *Naudotojo profiliai*. Taipogi galite paleisti QGIS su konkrečiu profiliu iš *komandinės eilutės*.

Patarimas

Paleiskite QGIS su nauju naudotojo profiliu, jei norite patikrinti rikto veikimą

Riktai, kuriuos aptinkate QGIS funkcijose, gali būti susiję su dabartiniame naudotojo profilyje likusiais dalykais. Paleidus QGIS su kitu naudotojo profiliu gali padėti juos išspręsti arba patikrinti problemos ištakas. Bendras patarimas yra paleisti QGIS su nauju naudotojo profiliu, taip gaunant švarią konfigūraciją, ir tada vėl vykdyti komandas.

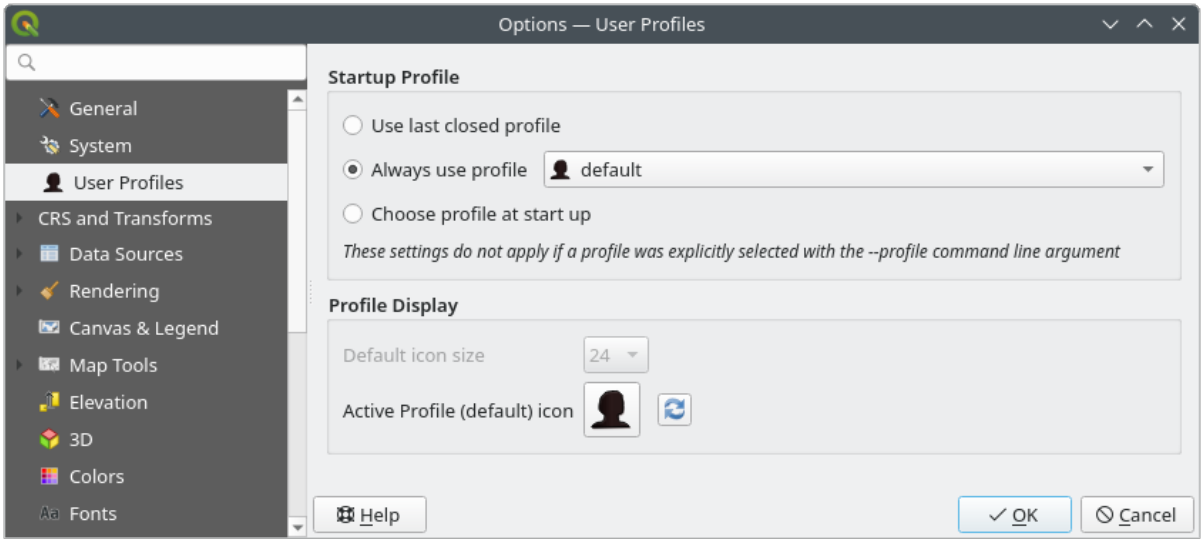
Jei riktas neleidžia jums sukurti naują naudotojo profilį per meniu *Nustatymai* ► *Naudotojo profiliai*, jūs galite:

- Failų naršyklėje pervadinti „sugedusį“ naudotojo profilį aplanke `QGIS3/profiles` ir iš naujo paleisti QGIS. Bus sukurtas ir įjungtas naujas `numatytasis` naudotojo profilis.
- Paleisti QGIS komandinėje eilutėje, naudojant naujo *profilio pavadinimo* argumentą:

```
qgis-ltr --profile newprofilename
```

5.2.2 Naudotojo profilio nustatymas

Pagal nutylėjimą QGIS atidaro naują sesiją su tuo profiliu, kuris uždarė paskutinę sesiją. Tai, kaip ir kitus nustatymus, galima keisti kortelėje *Nustatymai* ► *Parinktys* ► *Naudotojo profiliai*:



5.31 Fig. : Naudotojų profilių nustatymai

- *Pradinis profilis*: nurodo, kuris profilis parenkamas paleidžiant QGIS sesiją. Tai gali būti:
  - *Naudoti paskutinį užvertą profilį*
  - konkretus naudotojo profilis, parinktas iškrentančiame sąraše *Visada naudoti profilį*
  - *Parinkti profilį paleidimo metu*: Atveria dialogą *Naudotojo profilio parinkimas*, kuriame išvardinti visi naudotojų profiliai. Du kartus spauskite įrašą ar parinkite profilį ir spauskite *Gerai*, kad naudotumėte profilį pradedant sesiją. Taipogi galima *Pridėti naują profilį* į profilių sąrašą - jis bus sukurtas nuo nulio ir panaudotas atveriant sesiją.

- *Profilio rodymo* skiltyje jūs galite:
  - Nurodyti, kokio dydžio piktogramas naudoti parenkant profilį *Naudotojo profilių parinkimo* dialoge.
  - Parinkti konkrečią piktogramą, kuri turėtų būti rodoma greta dabartinio profilio meniu punkte *Nustatymai* ► *Naudotojo profiliai* ar *Naudotojo profilio parinkimo* dialoge. Spauskite Atstatyti piktogramą, kad atšauktumėte savo pakeitimus.

5.3 Projekto savybės

Projekto savybių lange *Projektas* ► *Savybės*, jūs galite nustatyti projekto parinktis. Projekto parinktys permuša jų ekvivalentus aukščiau aprašytame *Parinkčių* dialoge.

5.3.1 Bendros savybės

Kortelėje *Bendra*, *Bendri nustatymai* leidžia jums:

- pažiūrėti projekto failo vietą
- nustatyti projekto namų aplanką (matomas elemente *Projekto namai* skydelyje *Naršyklė*). Kelias gali būti santykinis nuo projekto failo (įrašykite jį) arba absoliutus. Projekto namuose gali būti laikomi duomenys ir kitas turinys, kuris naudingas projektui. Patogu, kai duomenų rinkinių ir projektų failai laikomi skirtingose vietose. Neužpildžius, *Projektų namai* bus aplankas, kuriame yra projekto failas.
- nustatyti projekto pavadinimą kartu su projekto failo keliu
- parinkti pažymėtų geoobjektų spalvą
- parinkti fono spalvą: žemėlapio drobės spalva
- nurodyti, ar keliai iki projekto sluoksnių turi būti įrašomi absoliutūs (pilni) ar santykiniai nuo projekto failo vietos. Jums gali labiau patikti santykiniai keliai, kai ir sluoksniai, ir projekto failai gali būti perkelti ir dalinami, arba jei projektas pasiekiamas iš kompiuterių su skirtingomis platformomis.
- pasirinkti vengti artefaktų, kai projektas braižomas kaip žemėlapio kaladėlės. Pastebėtina, kad įjungus šią parinktį gali sumažėti greitaveika.
- *Atsiminti atributų lentelių langus ir jų prisegimus tarp sesijų*: įjungus projektui šią parinktį, bet kuri atverta atributų lentelė bus įrašyta į projektą ir automatiškai įkelta atidarius projektą. Tai gali pagerinti darbų eigą, kai sukūrėte projektą su konkrečiomis atributų lentelės konfigūracijomis pagal jūsų reikalavimus ir iš naujo sudėlioti šias atributų lenteles užima daug laiko.

Plotų ir atstumų skaičiavimas yra dažnas GIS poreikis. Bet šios reikšmės tikrai susijusios su projekto nustatymais. *Matavimų skiltis leidžia valdyti šiuos parametrus. Jūs galite nurodyti:*

- *Elipsoidą*, kuriuo remiasi atstumo, ploto ir kampo skaičiavimai, jis gali būti:
  - **Jokio/Planimetriniai**: grąžinamos reikšmės yra dekarto matavimuose. Ši parinktis gali būti nustatyta kaip numatytoji naujiems projektams per meniu *Nustatymai* ► *Parinktys* ► *CRS valdymas*
  - **Savas**: jums reikės nurodyti semi-major ir semi-minor reikšmes.
  - arba esama iš sąrašo (Clarke 1866, Clarke 1880 IGN, Naujas tarptautinis 1967, WGS 84…).
- the *Units for distance measurements* for length and perimeter, and the *Units for area measurements*. These settings default to their corresponding *global options* and can be manually changed for the current project. They are used in:
  - Atributų lentelės lauko atnaujinimo juostoje
  - Laukų skaičiuotuvo skaičiavimuose
  - *Identifikavimo įrankio* išvestinio ilgio, perimetro ir ploto reikšmėms

  - *matavimo dialoge*
  - *mastelio dekoracijoje*

> **Pastaba**
>
> Siekiant užtikrinti, kad (atstumo ir ploto) matavimo vienetai visada atitiktų dabartinės projekto CRS vienetus, įjunkite šioms parinktims *Žemėlapio vienetai*.

- the *Scale calculation method*: when working on large map extents, especially with geographic CRS, the scale bar may not reflect actual distance measurement depending on the latitude at which the calculation is done. This setting helps you define the area you want the calculation to be taken from:
  - *Average top, middle and bottom scales*
  - *Calculate along top of map*
  - *Calculate along middle of map* (the default)
  - *Calculate along bottom of map*
  - *Always calculate at equator*, regardless of the actual visible map extent. It can be used to provide a consistent, static scale for maps in geographic reference systems, regardless of the latitudes actually visible in the map (permitting consistent appearance of these maps when rendering relies on scale based visibility or calculations). Otherwise a project in e.g. `EPSG:4326` which uses scale based visibility of layers and symbols will see layers and features „randomly“ disappear as the map is panned, even though you have not zoomed in or out of the map.

    > **Pastaba**
    >
    > This method is only applicable when calculating scales with a degree based reference system, and while it ensures that the scale remains constant and does not change as the map is panned, it will calculate misleading scales when the map extent is not close to the equator.

  Changing the scale calculation method has the following impacts:
  - New layout scale bars will default to the project's scale calculation method
  - The scale calculations for the status bar widget and map renders will be changed (including flow on impacts like the value of `@map_scale` variable, scale based visibility of layers and symbols)
  - This also affects QGIS server map rendering
  - Processing algorithms which render maps will respect the project's scale calculation method

> **Pastaba**
>
> Symbology sizes in map units are NOT affected by this setting.

Koordinačių ir krypties rodymas leidžia valdyti, kaip vaizduojamos:

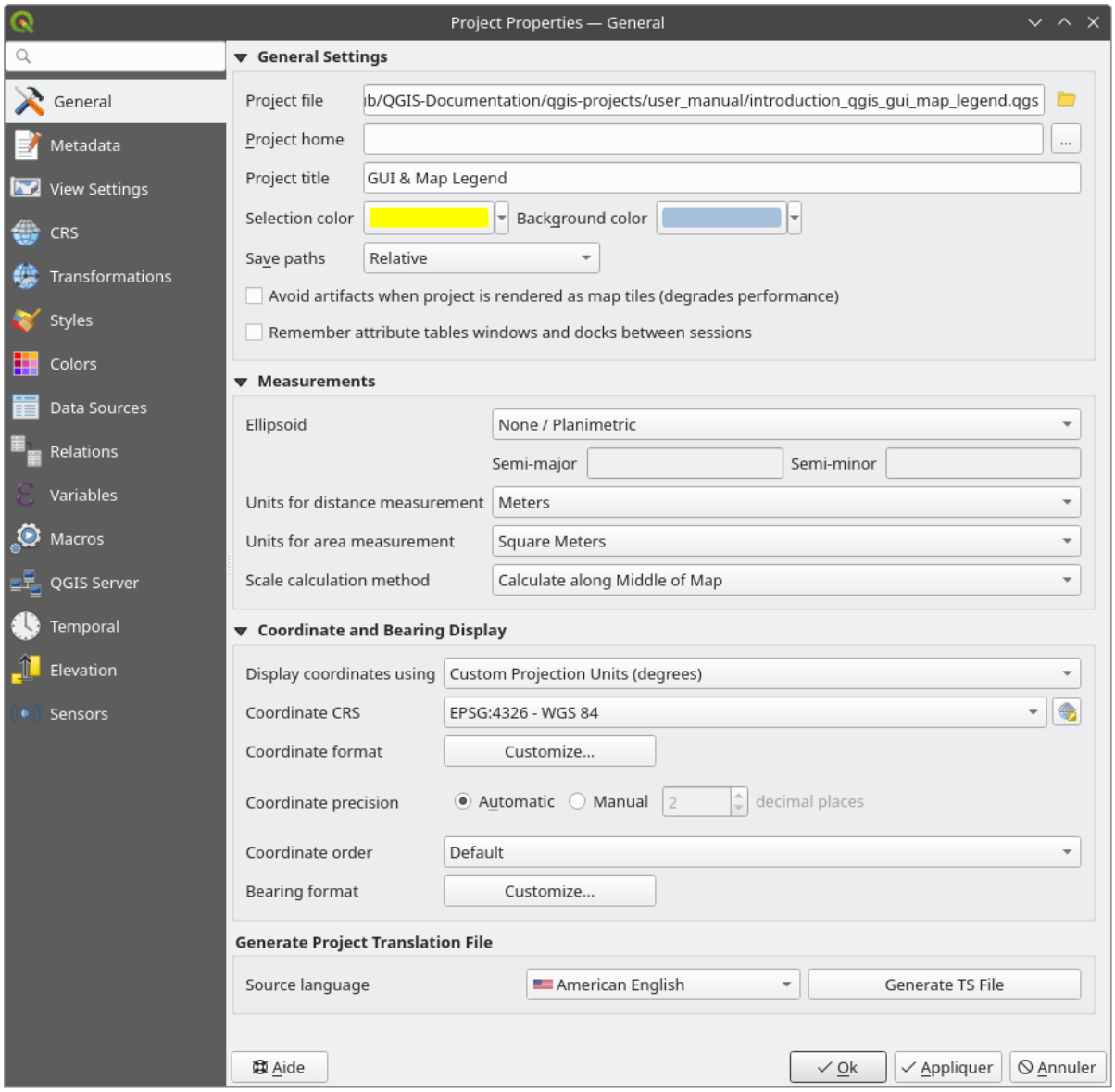
- koordinatės *Koordinačių* langelyje QGIS būsenos juostoje ir Geoobjektų identifikavimo rezultatų skiltyje *Išvestinė*
- krypties reikšmė būsenos juostoje rodant žemėlapio drobės slinkimo kryptį ir Matavimo krypties įrankio.

Galimi parametrai:

- *Koordinačių atvaizdavimo vienetai*:
  - `Žemėlapio vienetai` pagal projekto CRS
  - `Geografinės (laipsniai)`: pagal projekto CRS, jei geografinio tipo, priešingu atveju naudoja susietą geografinę CRS. Tai naudinga pvz. ne Žemės kosminiams kūnams.

  - arba `Savos projekcijos vienetai`: leidžia koordinačių rodymui remtis bet kuria jums reikiama CRS

- *Koordinačių CRS* parinktis leidžia žiūrėti ar nurodyti CRS, kurią reikia naudoti priklausomai nuo rodymo režimo.
- *Koordinačių formatas*: galite sukonfigūruoti jį kaip `Dešimtainiai laipsniai`, `Laipsniai minutės` arba `Laipsniai, Minutės, Sekundės` ir ar reikėtų:
  - ☐ *Rodyti krypties priesagą*
  - ☐ *Rodyti nulius minučių ir sekundžių priekyje*
  - ☐ *Rodyti nulius laipsnių priekyje*
  - ☐ *Rodyti nulius pabaigoje*
- *Koordinačių tikslumas*: skaitmenų po kablelio skaičius gali būti automatinis (gaunamas pagal CRS tipą) arba nurodomas rankiniu būdu
- *Koordinačių tvarka*: jūs galite pasirinkti rodyti koordinates CRS tvarka (`Numatytoji`) arba persijungti į `Rytai, Šiaurė (Ilguma, Platuma)` arba `Šiaurė, Rytai (Platuma, Ilguma)` tvarką
- *Krypties formato* galimos reikšmės yra `0 iki 180°, su E/W priesaga`, `−180 iki +180°` arba `0 iki 360°`. Taipogi galima nurodyti *Skaitmenų po kablelio* skaičių bei ar *Rodyti nulius pabaigoje*.

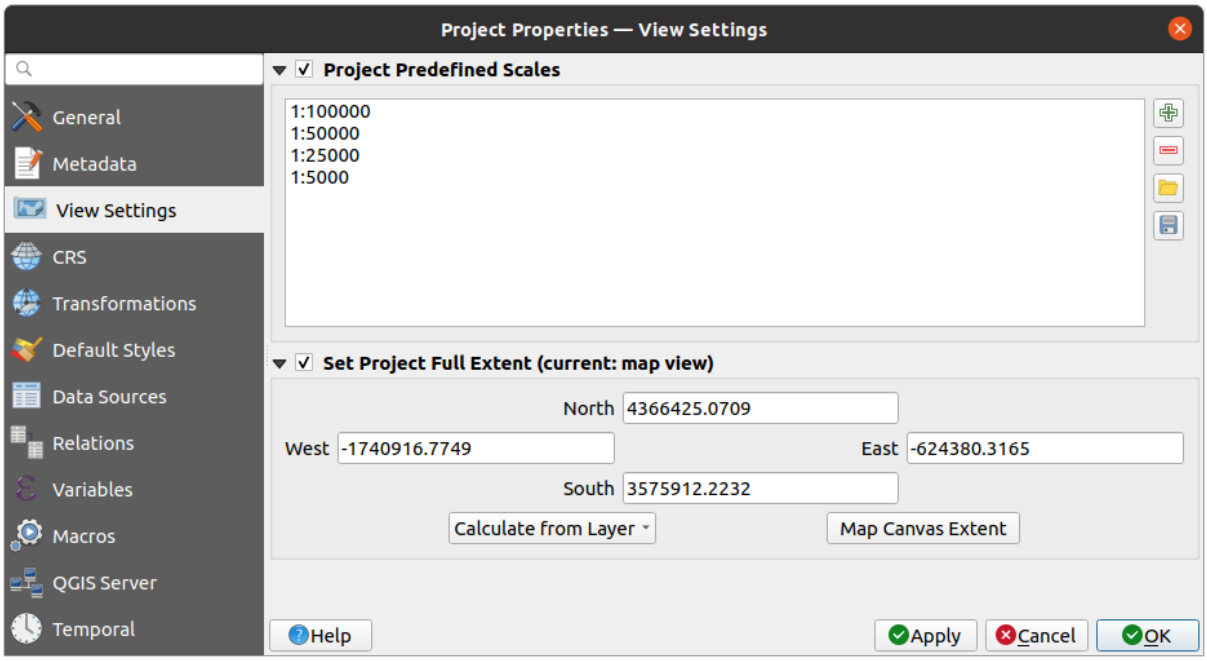


5.32 Fig. : Projekto savybių dialogo Bendra kortelė

5.3.2 Metaduomenų savybės

The *Metadata* tab provides you with options to create and edit a metadata report on your project. See *Metaduomenys* for more information.

5.3.3 Vaizdo nustatymai



5.33 Fig. : Projekto savybių dialogo vaizdo nustatymo kortelė

Kortelė *Vaizdo nustatymai* leidžia valdyti projekto žemėlapio drobę. Jūs galite:

- nustatyti *Iš anksto apibrėžti projekto mastelius*: mastelių sąrašas, rodomas su masteliais susijusiuose iškrentančių sąrašų valdikliuose, tokiuose kaip būsenos juostos *Mastelis*, matomumo masteliai arba antrinio 2D žemėlapio vaizdo nustatymai… vietoje globalų *iš anksto numatytų mastelių*.
- *Nustatyti pilną projekto apimtį*: ši apimtis bus naudojama vietoje visų sluoksnių apimties didinant iki pilno žemėlapio apimties ( ). Tai naudinga, kai projekte yra tinklo sluoksnių, nacionalinių duomenų rinkinių, globalių sluoksnių, nors reali projekto dominanti teritorija yra mažesnė. Projekto pilnos apimties koordinates galima nustatyti naudojant *apimties parinkimo* valdiklį.

5.3.4 CRS savybės

Pastaba

Daugiau informacijos apie tai, kaip QGIS tvarko projekto projekcijas, rasite skyriuje *Darbas su projekcijomis*.

Kortelė *CRS* leidžia nustatyti projekte naudojamą koordinačių atskaitos sistemą. Ji gali būti:

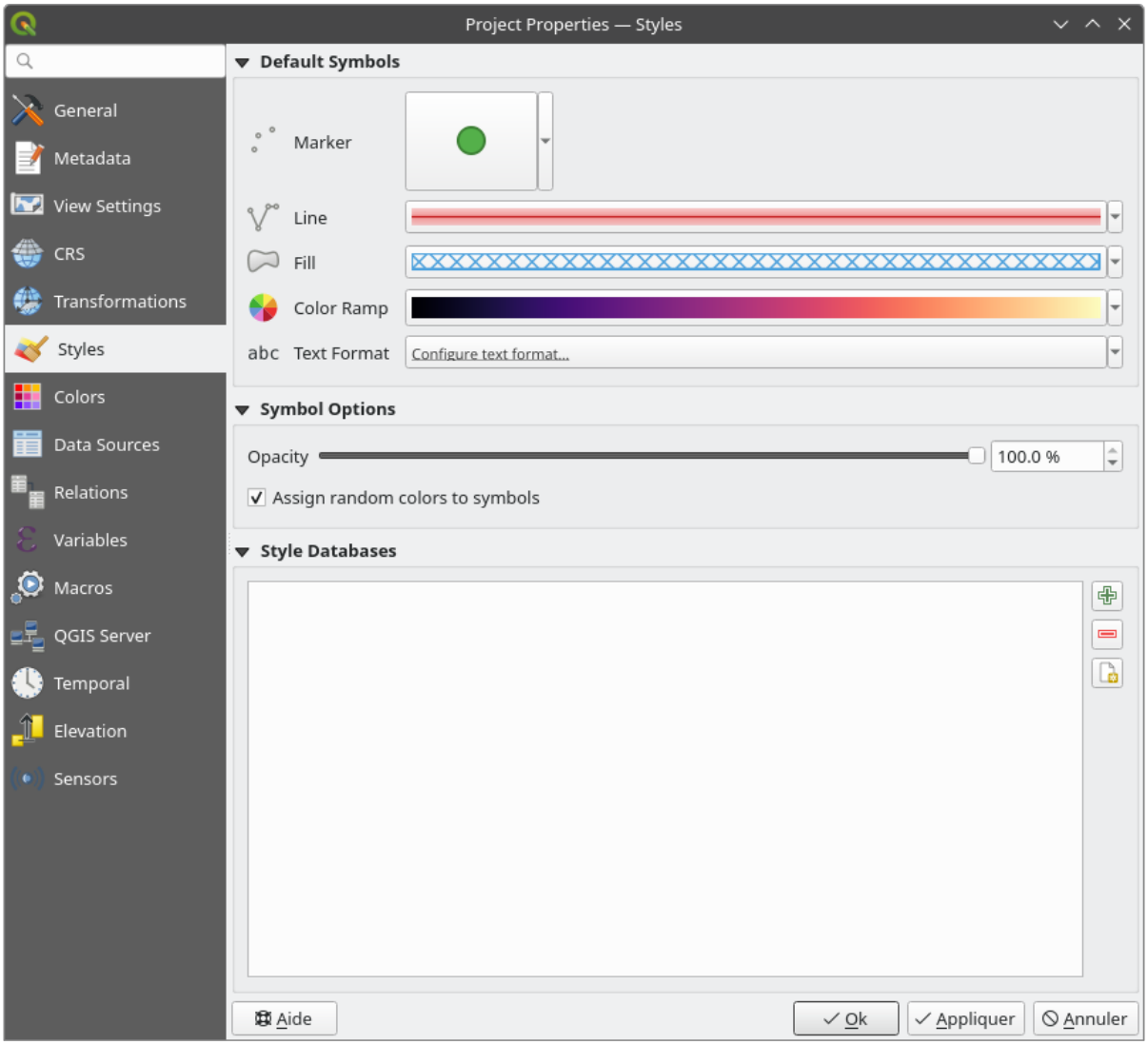
- *Be CRS (arba nežinoma/ne-Žemės projekcija)*: sluoksniai braižomi pagal savo pradines koordinates
- arba esama koordinačių atskaitos sistema, kuri gali būti *geografinė*, *projektuota* arba *naudotojo-nustatyta*. Į projektą pridedami sluoksniai automatiškai keičiami į šią CRS nepriklausomai nuo pradinės CRS.

5.3.5 Transformacijų savybės

Kortelė *Transformacijos* leidžia valdyti sluoksnių perprojektavimo nustatymus, konfigūruojant duomenų transformacijos nuostatas, taikomas dabartiniam projektui. Kaip įprasta, tai permuša atitinkamus globalius nustatymus. Daugiau informacijos rasite *Duomenų transformacija*.

5.3.6 Stilių savybės

Kortelėje *Stiliai* jūs galite konfigūruoti į projektą įrašytus simbolius ir spalvas, taip leidžiant saugiai dalintis projektu tarp skirtingų mašinų.



5.34 Fig. : Stilių kortelė

Numatytieji simboliai

Grupė *Numatytieji simboliai* leidžia valdyti, kaip bus braižomi nauji sluoksniai, kai jie neturi apibrėžto susijusio `.qml` stiliaus. Jūs galite nustatyti *Žymeklį*, *Liniją*, *Užpildymą*, kurie bus taikomi priklausomai nuo sluoksnio geometrijos tipo, bei numatytąją *Spalvų rampą* ir *Teksto formatą* (pvz. naudojant užrašus). Bet kurį iš šių elementų galima atstatyti atitinkamo valdiklio iškrentančiame meniu parinkus *Išvalyti*.

Simbolių parinktys

Simbolių parinkčių grupėje jūs galite:

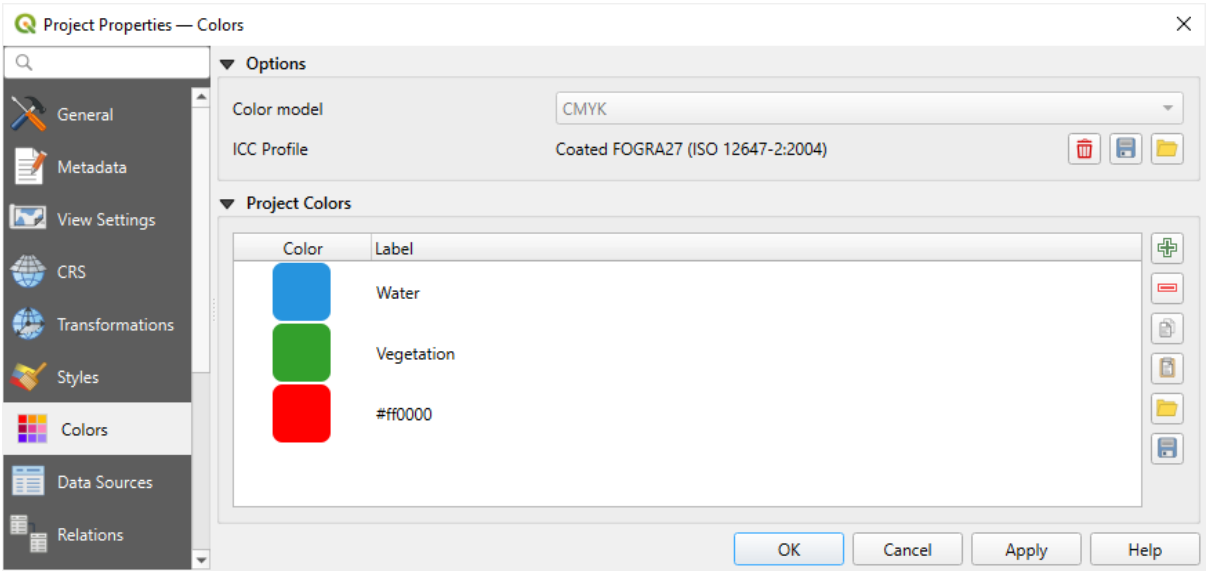
- Pritaikyti numatytąjį *Nepermatomumą* naujiems sluoksniams
- *Priskirti simboliams atsitiktines spalvas*, keičiant simbolių pildymo spalvas, taip išvengiant identiško visų sluoksnių braižymo.

Stilių duomenų bazė

QGIS stilių duomenų bazė - tai struktūrinė repozitorija, kurioje saugomi simboliai, teksto formatai ir kiti stiliaus elementai. Ji veikia kaip centralizuota vieta, kurioje jūs galite efektyviai organizuoti ir valdyti jūsų simbolizacijos resursus. Jūs galite sukurti konkrečiam klientui skirtą stilių duomenų baze, namų simbolius, pritaikytus būtent to kliento poreikiams. Tai užtikrins švarų ir organizuotą būdą, neapkraunant numatytųjų stilių duomenų bazės. Kelių naudotojų aplinkose galima laikyti konkrečiam projektui pritaikytus stilius bendrinamoje vietoje. Susiejus projektą su šiais stiliais, visi projekto naudotojai gauna prieigą prie bendros simbologijos, taip išvengiant poreikio rankiniu būdu importuoti ir atnaujinti savo individualią stilių duomenų bazę. Simbolių saugojimas projekto faile suteikia praktinį sprendimą, leidžiantį naudotojams neapkrauti savo globalios stilių duomenų bazės su tik konkrečiam projektui naudojama simbologija. Šis būdas garantuoja, kad kiti to paties projekto naudotojai įsikėlę projektą nedelsiant turi prieigą prie visos reikiamos simbologijos.

Skiltyje *Stilių duombazės* jūs galite pasirinkti *Pridėti* ar *Pašalinti* stilių duombazę bei galite *Sukurti naują stilių duombazę*. Kai šioje skiltyje pridedate ar išimate stilių duombazę, pakeitimai automatiškai atsispindės *Simbologijos savybėse*.

5.3.7 Spalvų savybės



5.35 Fig. : Spalvų kortelė

Kortelėje *Spalvos* jūs galite parinkti *Spalvų modelį*, t.y. ar pagal nutylėjimą spalva parenkama kaip `RGB`, ar `CMYK`. Bet kokia spalva, apibrėžta kitame nei nurodytame spalvų modelyje, bus pakeista į nurodytą modelį eksportuojant išdėstymą.

Additionally, an option is available for selecting an *ICC Profile* which allows you to *Load* a specific ICC profile that defines the color space for your project. After you load the ICC profile, you can choose to *Save ICC Profile* to a file on disk or to *Remove selected ICC Profile*. Once applied, QGIS will enforce the selected color model to ensure consistent color representation.

Pastaba

Note that the *ICC Profile* option will only be visible if QGIS is built with Qt 6.8.0 or higher.

Taipogi yra papildoma skiltis, kurioje jūs galite nurodyti konkrečias spalvas einamajam projektui. Kaip ir *globaliose spalvose*, jūs galite:

- *Pridėti* ar *Išimti* spalvą
- *Kopijuoti* ar *Įkelti* spalvą
- *Importuoti* ar *Eksportuoti* spalvų rinkinį iš/į `.gpl` failą.

Du kartus paspaudus ant spalvos sąraše, galite ją tvarkyti ar pakeisti *Spalvų parinkimo* dialoge. Taip pat galite ją pervadinti du kartus paspaudę ant *Etiketės* stulpelio.

Šios spalvos vadinamos *Projekto spalvomis* ir įtraukiamos į *spalvų valdiklių* sąrašus.

Patarimas

Naudokite projekto spalvas, kai norite greitai priskirti ar pakeisti spalvų valdiklius

Projekto spalvas galima naudoti pagal jų pavadinimą, kurį atpažins visi spalvų valdikliai. Tai reiškia, kad vietoje to, kad pakartotinai nustatytumėte tą pačią spalvą daugeliui savybių, jūs galite:

1. Apibrėžkite spalvą kaip projekto spalvą
2. Spauskite *permušimo pagal duomenis valdiklį*, esantį greta spalvos, kurią norite nustatyti
3. Užveskite pelę virš *Spalvų* meniu ir parinkite projekto spalvą. Savybei tada bus priskirta išraiška `project_color('spalvos-pavadinimas')` ir spalvos valdiklis atspindės tą spalvą.
4. Pakartokite 2 ir 3 žingsnius tiek kartų, kiek reikia
5. Pakeiskite projekto spalvą vieną kartą ir pakeitimas atsispindės VISUR, kur naudojama ši spalva.

5.3.8 Duomenų šaltinių savybės

Kortelėje *Duomenų šaltiniai* jūs galite:

- *Transakcijų režimas*, nurodyti, kaip pakeitimai siunčiami duomenų tiekėjui:
  - *Vietinis redagavimo buferis*: pakeitimai laikomi vietiniame buferyje ir siunčiami tiekėjui, kai keičiamas sluoksnio redagavimo režimas arba paspaudžiamas mygtukas *Įrašyti pakeitimus*.
  - *Automatinės transakcijų grupės*: tai palaikančiuose duomenų šaltiniuose (postgres ir geopackage duomenų bazėse) visų tos pačios duomenų bazės lentelių redagavimo būsena sinchronizuojama ir vykdoma transakcijoje serveryje. Taipogi, vietoje vietinio pakeitimų saugojimo, jei siunčiami tiesiai į transakciją

duomenų bazėje, kur jie patvirtinami keičiant sluoksnio redagavimo režimą arba spaudžiant mygtuką *Įrašyti pakeitimus*.

- *Buferinės transakcijų grupės*: visi redaguojami sluoksniai, nepriklausomai nuo jų tiekėjo, sinchroniškai perjungiami ir visi pakeitimai įrašomi į vietinį keitimų buferį. Pakeitimų įrašymas vykdomas viena transakcija visiems sluoksniams (per duomenų tiekėją).

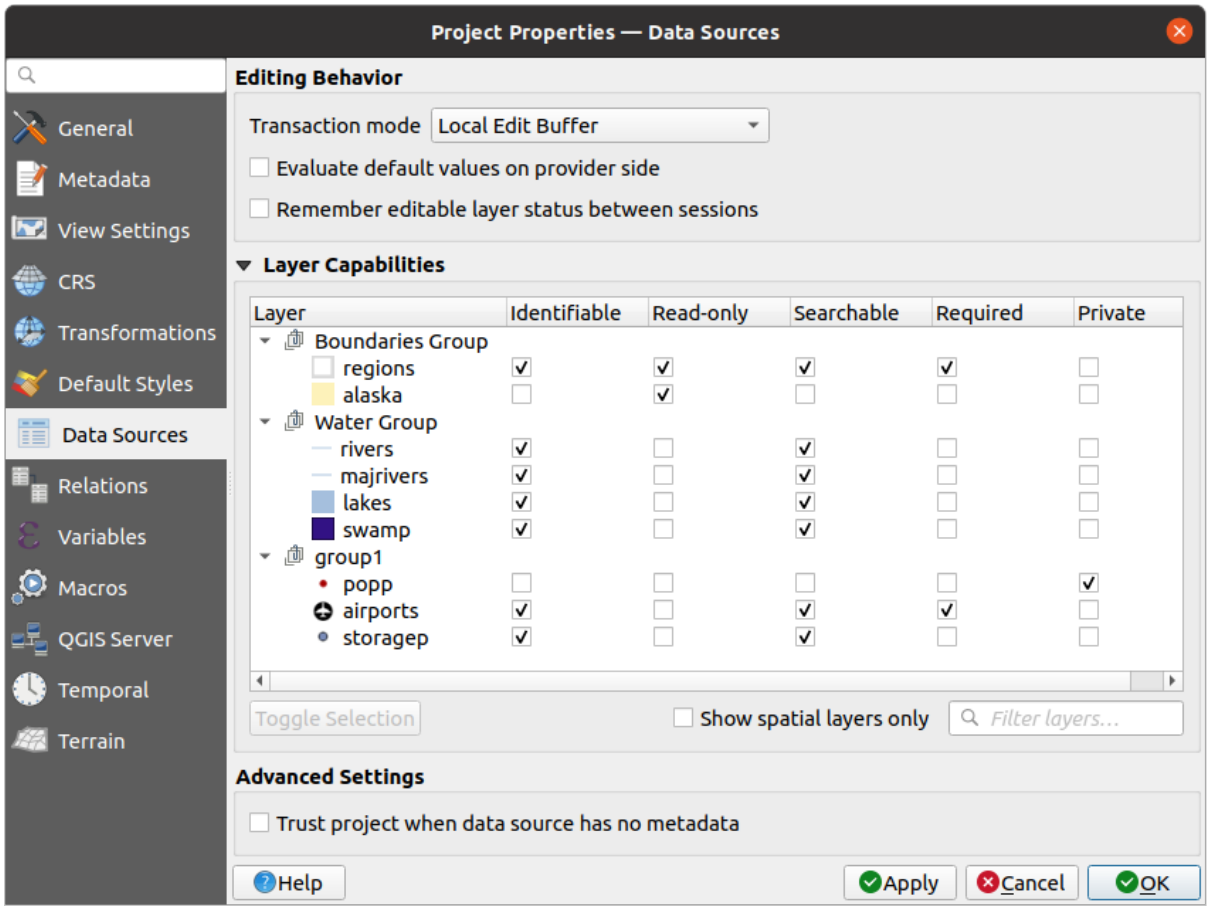
Pastebėtina, kad šią parinktį keisti galite tik jei projekte nėra redaguojamas nei vienas sluoksnis.

- *Įvertinti numatytąsias reikšmes tiekėjo pusėje*: Pridedant naujas reikšmes į PostgreSQL lentelę, laukai su numatytosios reikšmės apribojimu įvertinami ir užpildomi atidarant formą, o ne patvirtinant pakeitimus. Tai reiškia, kad vietoje išraiškos, tokios kaip `nextval('serial')`, lauke *Pridėti geoobjektą* formoje bus rodoma tikra reikšmė (pvz., `25`).
- *Atsiminti redaguojamo sluoksnio būseną tarp sesijų*: užtikrina, kad visi sluoksniai, kurie yra redaguojami, bus atsiminti kaip tokie įrašant projektą. Atidarius projektą šie sluoksniai nedelsiant bus perjungti į redagavimo režimą.
- Konfigūruoti *Sluoksnių galimybes*, t.y.:
  - Nustatyti (ar išjungti), kurie sluoksniai yra `identifikuojami`, t.y. reaguos į *identifikavimo įrankį*. Pagal nutylėjimą, sluoksniai atsako į užklausas.
  - Nustatyti, ar sluoksniai turi būti `tik skaitymo`, tai reiškia, kad naudotojas jų negali redaguoti, nepriklausomai nuo to, ar duomenų tiekėjas tai leidžia. Nors tai ir silpna apsauga, tai yra greita ir lengva konfigūracija, leidžianti išvengti situacijų, kai naudotojai keičia duomenis dirbdami su failais paremtais sluoksniais.
  - Nustatyti, kurie sluoksniai yra `ieškomi`, t.y. kuriuose galima vykdyti užklausas naudojant *lokatoriaus valdiklį*. Pagal nutylėjimą sluoksniuose galima ieškoti.
  - Nustatyti, kurie sluoksniai yra `privalomi`. Tokie sluoksniai apsaugoti nuo netyčinio išėmimo iš projekto.
  - Nustatyti, kurie sluoksniai yra `privatūs`, t.y. paslėpti iš *Sluoksnių* skydelio. Tai skirta pagalbiniams sluoksniams (baziniams sluoksniams, jungtims, reikšmių ryšiams, klasifikatoriams, tikėtina sluoksniams be geometrijų …), kurie projekte reikalingi, bet jūs nenorite, kad jie užterštų legendą ir kitus sluoksnio parinkimo įrankius. Nustačius kaip matomus, jie vis tiek bus rodomi žemėlapio drobėje ir braižomi spausdinimo maketų legendoje. Naudokite parinktį *Filtruoti legendą* ► *Rodyti privačius sluoksnius*, esančią *Sluoksnių* skydelio viršutinėje įrankinėje, jei norite laikinai juos įjungti kokiam nors veiksmui.

Lentelės *Sluoksnių galimybės* įrankiai leidžia patogiai:

- Pažymėti kelis langelius ir spausti *Perjungti pažymėjimą*, kad pakeistumėte jų varnelių būseną;
- *Rodyti tik erdvinius sluoksnius*, filtruoti neerdvinius sluoksnius iš sąrašo;
- *Filtruoti sluoksnius…* ir greitai rasti konfigūravimui konkretų sluoksnį.

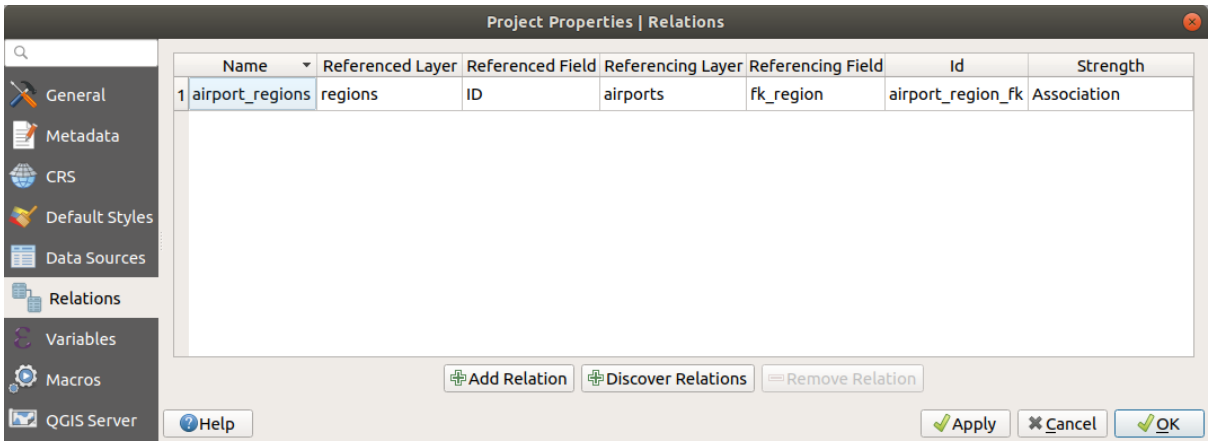
- *Išmanesnių nustatymų* grupėje jūs galite įjungti *Pasitikėti projektu, kai duomenų šaltinis neturi metaduomenų*: taip pagreitinsite projekto įkėlimą praleisdami duomenų tikrinimus. Naudinga QGIS Serverio kontekste arba projektuose su didžiuliais duomenų vaizdais/materializuotais vaizdais. Sluoksnių apimtys bus skaitomos iš QGIS projekto failo (o ne iš duomenų šaltinio), kai naudojamas PostgreSQL tiekėjas, vaizdams ir materializuotiems vaizdams nebus tikrinamas pirminio rakto unikalumas.



5.36 Fig. : Duomenų šaltinių kortelė

5.3.9 Ryšių savybės

Kortelėje *Ryšiai* galima apibrėžti ryšius tarp lentelių. Ryšiai gali būti vienas su vienu, daug su daug, ar polimorfinių tipų. Juos galima apibrėžti projekto savybių dialoge. Kai sluoksnis turi ryšių, naujas naudotojo sąsajos elementas formos vaizde (pvz. identifikuojant geoobjektą ar atidarius jo formą) pateiks susijusių esybių sąrašą. Tai teikia galingą būdą išreikšti pvz. vamzdžių ilgio ar kelio segmento tikrinimų istoriją. Daugiau apie ryšių palaikymą galite rasti skiltyje *Setting relations between multiple layers*.



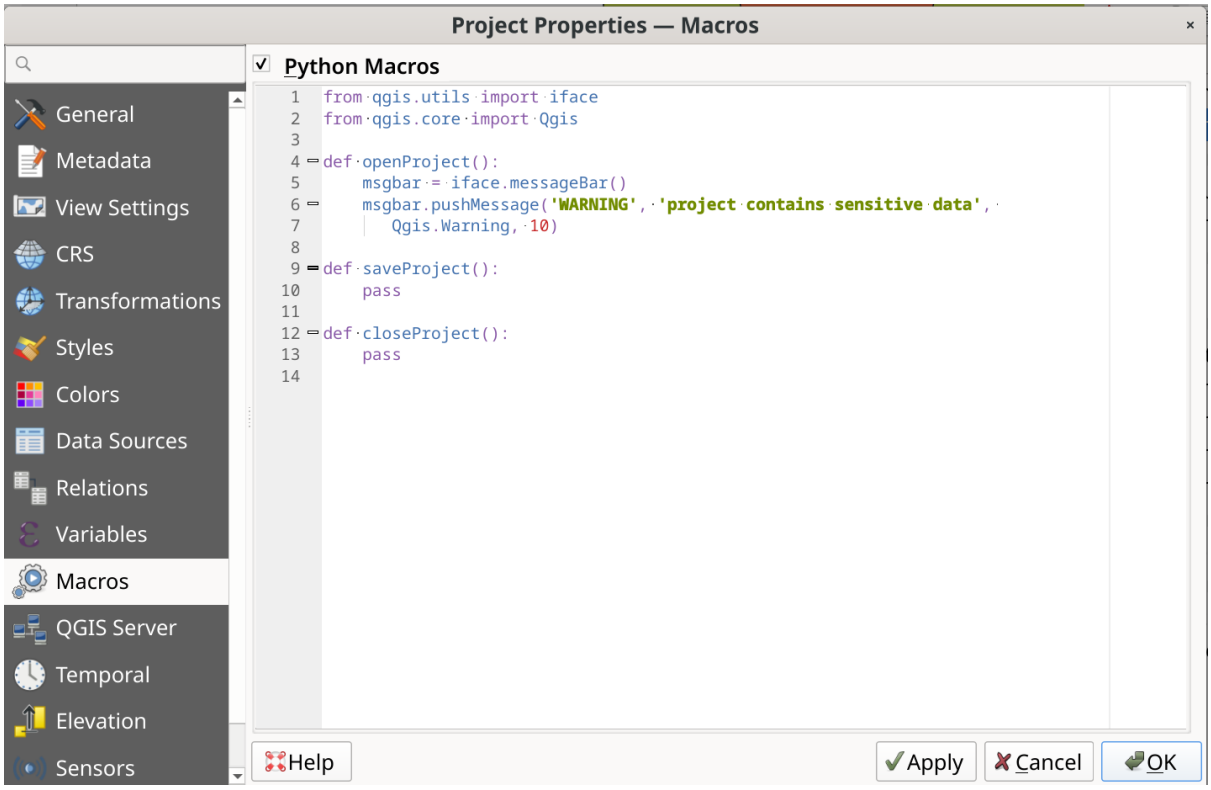
5.37 Fig. : Ryšių kortelė

5.3.10 Kintamųjų savybės

Kortelėje *Kintamieji* išvardinti projekto lygio kintamieji (įskaitant visus globalius kintamuosius). Be to, naudotojas čia gali tvarkyti projekto lygio kintamuosius. Spauskite mygtuką , jei norite pridėti naują projekto lygio kintamąjį. Analogiškai, parinkite savo projekto lygio kintamąjį sąrašą ir spauskite mygtuką , kad jį panaikintumėte. Daugiau informacijos apie kintamųjų naudojimą rasite Bendrųjų įrankių skiltyje *Reikšmių saugojimas kintamuosiuose*.

5.3.11 Makro savybės

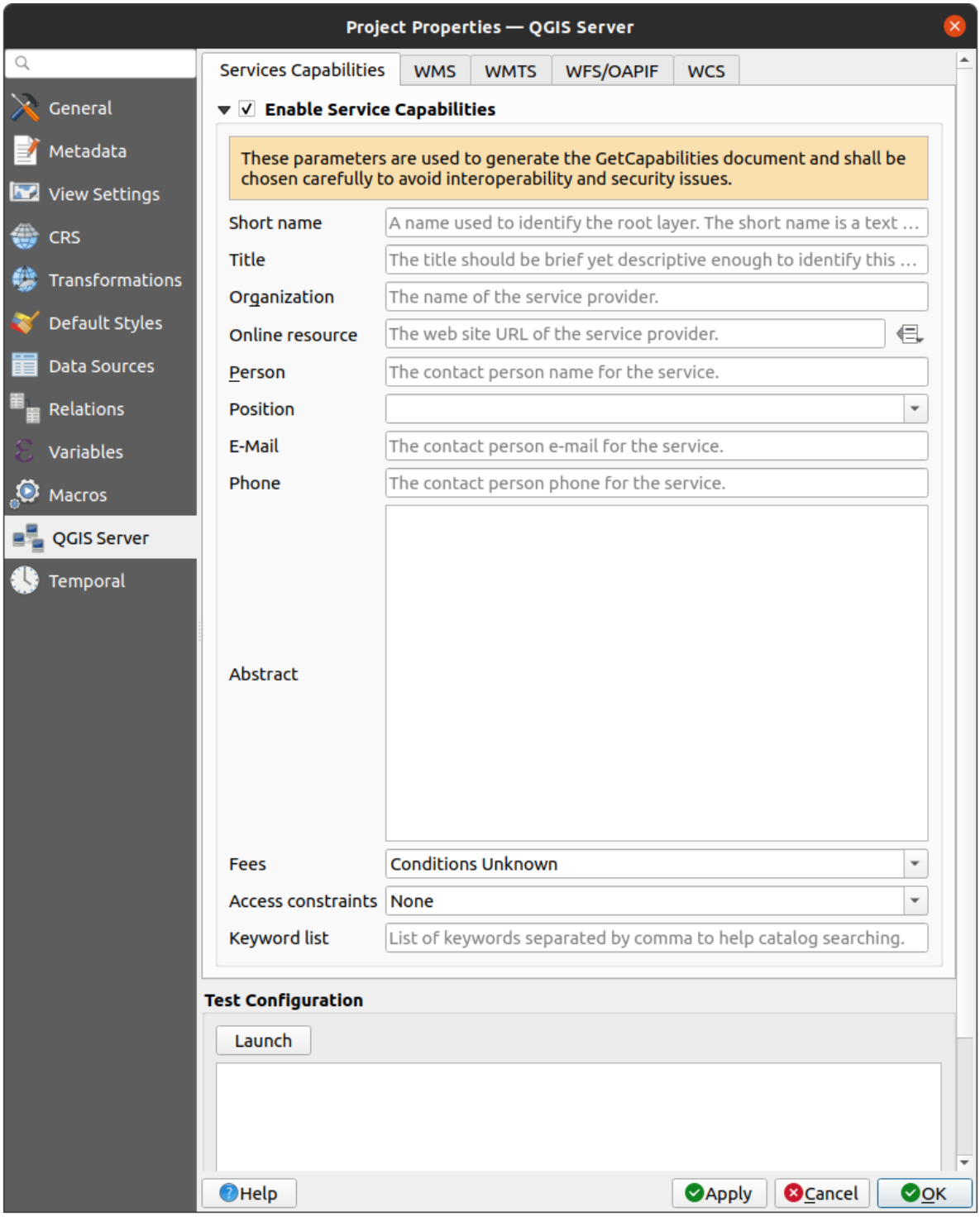
Kortelė *Makro* leidžia keisti projekto Python makro. Šiuo metu galimi tik trys makro: `openProject()`, `saveProject()` ir `closeProject()`.



5.38 Fig. : Makro nustatymai

5.3.12 QGIS serverio savybės

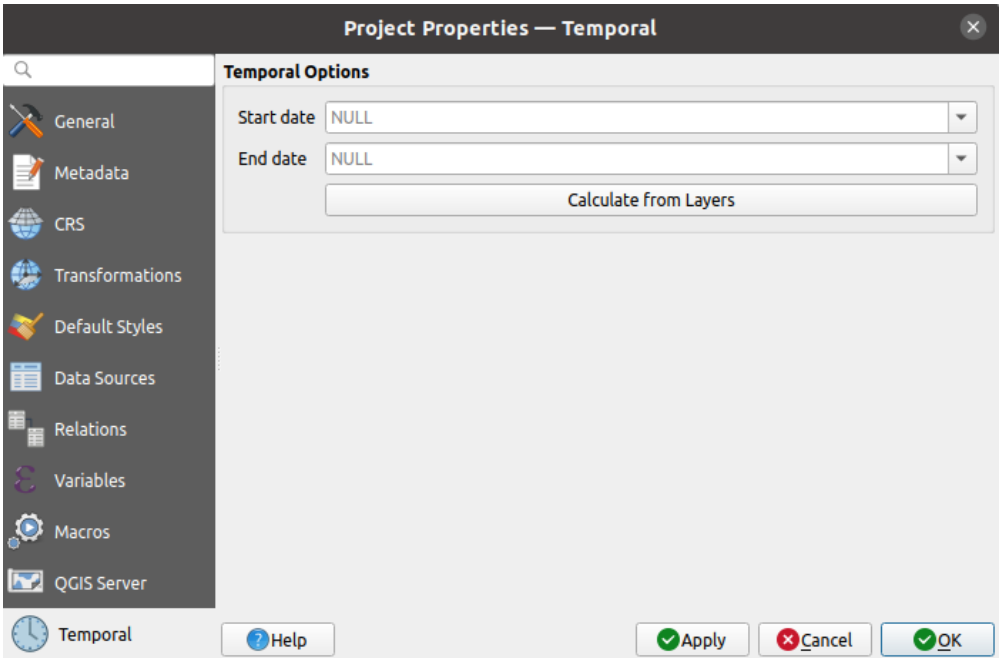
Kortelėje *QGIS Serveris* galima konfigūruoti jūsų projektą, jei norite jį publikuoti internete. Čia jūs galite nustatyti informaciją apie QGIS Serverio WMS ir WFS galimybes, apimtį ir CRS apribojimus. Daugiau informacijos rasite skiltyje Creatingwmsfromproject ir toliau.



5.39 Fig. : QGIS serverio nustatymai

5.3.13 Laiko savybės

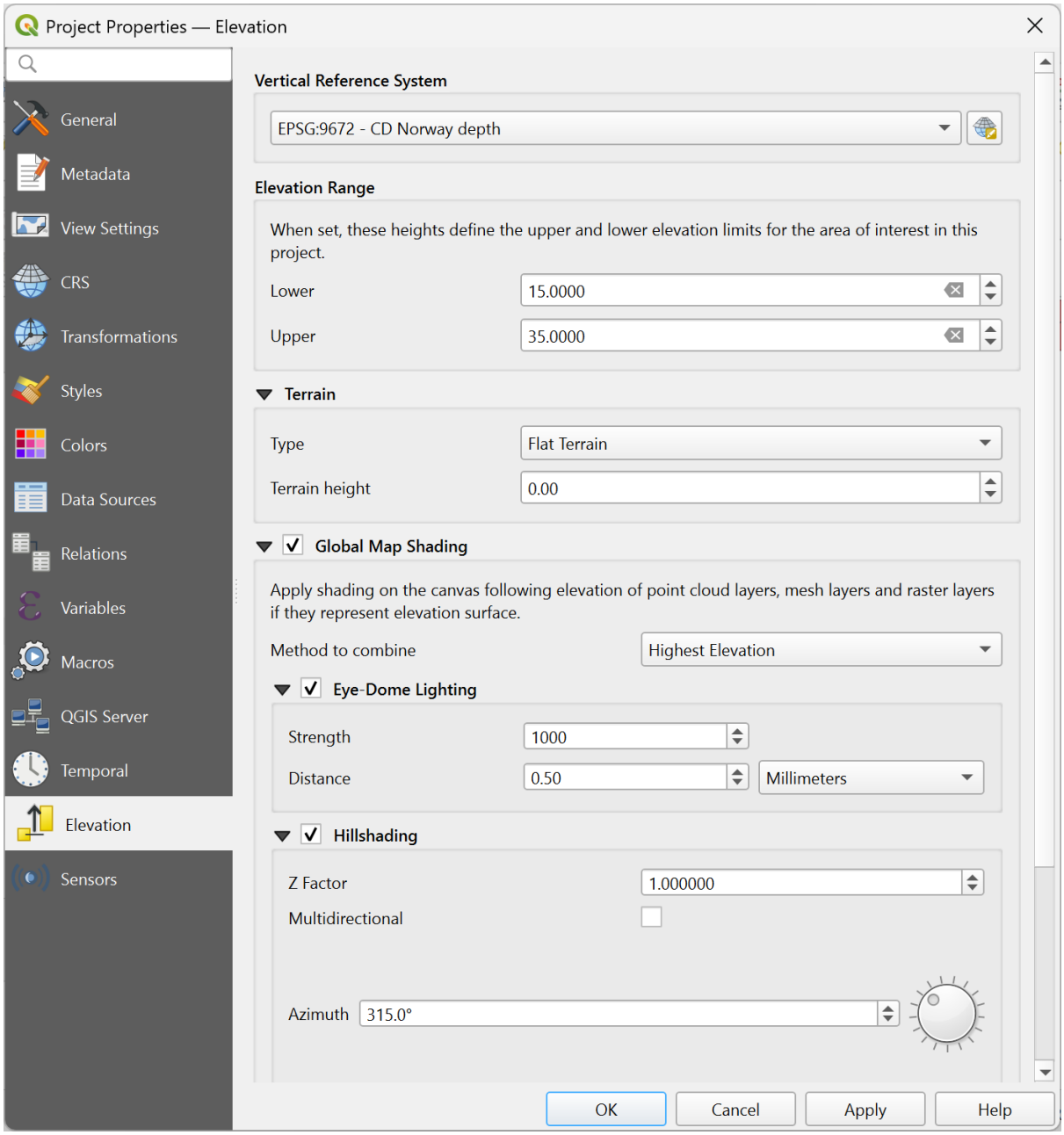
Kortelė *Laikas* skirta nustatyti jūsų projekto laiko diapazoną arba naudojant rankinę *Pradžios datą* ir *Pabaigos datą*, arba skaičiuojant jį pagal dabartinio projekto su laiku susietus sluoksnius. Projekto laiko diapazonas tada gali būti naudojamas *Laiko valdymo skydelyje*, norint valdyti žemėlapio drobės *navigaciją laike*.



5.40 Fig. : Projekto laiko kortelė

5.3.14 Aukščio savybės

Kortelė *Aukštis* leidžia jums konfigūruoti numatytuosius paviršiaus ir aukščio nustatymus. Kai projekte sukuriamas naujas *3d žemėlapis*, jis naudos tokius paviršiaus nustatymus, kokie apibrėžti projekte. Įrankis *aukščio profilis* taipogi atsižvelgs ir į projekto aukščio nustatymus.



5.41 Fig. : Projekto aukščio kortelė

- *Vertikali atskaitos sistema*: Jei jūsų projekto CRS yra sudėtinis (turi ir Z matmenį), tada projekte naudojama vertikali CRS bus projekto CRS vertikalus komponentas. Jei projekto CRS yra horizontali, tada jūs galite parinkti konkrečią vertikalią CRS, paspausdami Parinkti CRS.
- *Aukščio diapazonas*: padeda jums nustatyti *Apatinį* ir *Viršutinį* apribojimus *aukščio kontroleriui* projekte.
- *Paviršiaus* ir aukščio parinktys galimos:
  - *Plokščiam paviršiui* su *Paviršiaus aukščio* nustatymu
  - *DEM (Rastro sluoksniui)*: su nustatymu, apibrėžiančiu *Rastro sluoksnį*, *Vertikalaus mastelio* faktorių, taikomą juostų reikšmėms ir vertikalų *Poslinkį*
  - *Tinkleliui*: su nustatymais, apibrėžiančiais *Tinklelio sluoksnį*, *Vertikalaus mastelio* faktorių, taikomą viršūnių Z reikšmei ir vertikalų *Poslinkį*

Šiuos nustatymus galima permušti 3D žemėlapio *konfigūravimo dialoge*.

- *Globalaus žemėlapio šešėliavimo* nustatymai leidžia nurodyti globalius žemėlapio šešėliavimo efektus, kurie priklauso nuo visų aktyvių sluoksnių, turinčių aukščio atributą, aukščio, įskaitant:
  - *rastro sluoksnius*
  - *tinklelio sluoksnius*
  - *taškų masyvo sluoksnius*

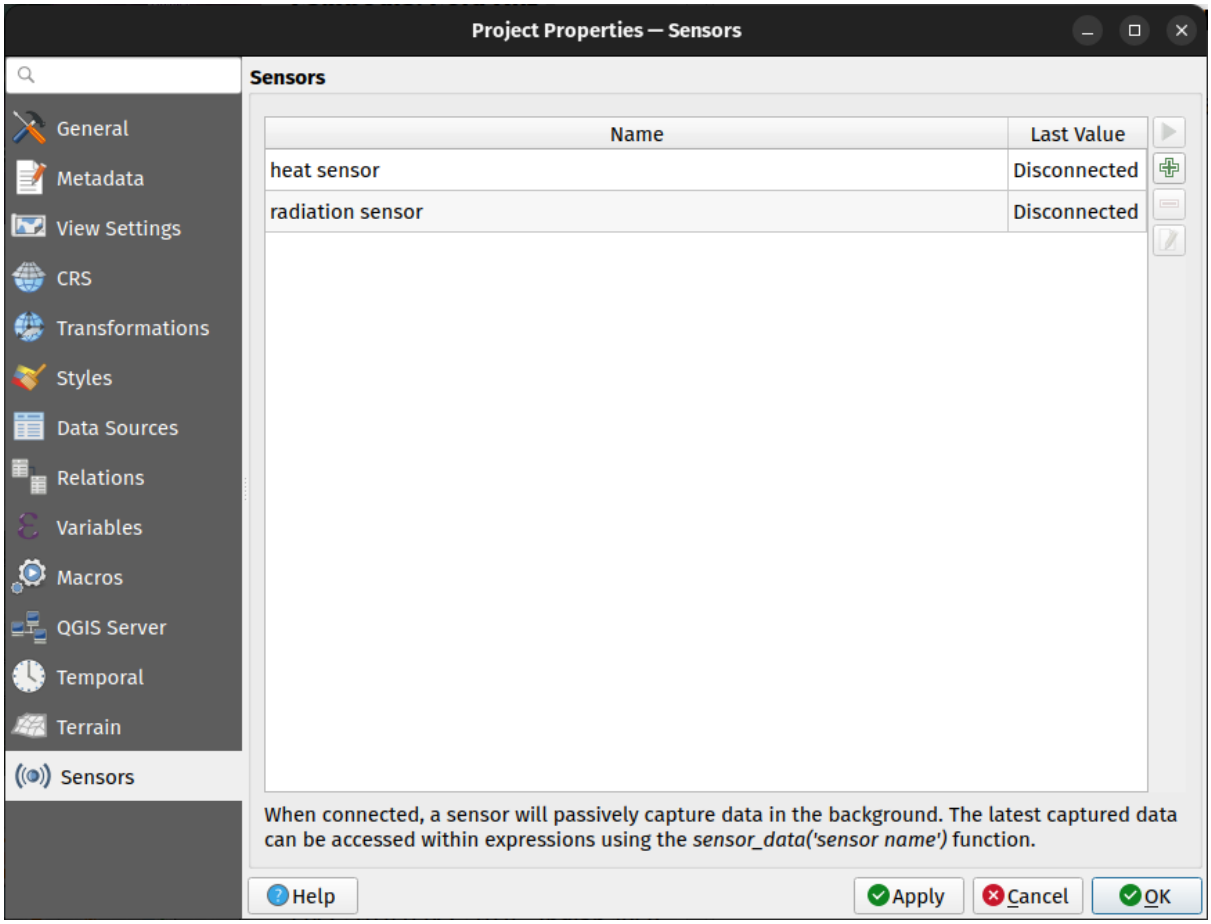
  Kombinavimo būdas: kiekvienos konkrečios vietos aukščio reikšmė iš visų sluoksnių yra kombinuojama, ir vertinamas aukštis pasirenkamas pagal vieną iš šių būdų:
  - *Didžiausio aukščio* reikšmė
  - *Pagal sluoksnių rikiuotę*: vertinamas aukščiausiai esančio sluoksnio aukštis.

  Palaikomos šios šešėliavimo parinktys:
  - *Akių kupolo apšvietimas*: taiko žemėlapio drobei šešėliavimo efektus geresniam gylio vaizdavimui. Galima valdyti šiuos parametrus:
    - *Stiprumas*: didina kontrastą, taip leidžiant geresnį gylio suvokimą
    - *Atstumas*: reprezentuoja naudotų taškų atstumą nuo taško centro ir padaro kraštus storesniais.
  - *Šešėliavimas*, kai kurių žemėlapio reljefų formavimas naudojant šešėliavimą (pilkos lygius):
    - *Z faktorius*: Aukščio reikšmės daugiklis
    - *Kelių krypčių*: Nurodykite, jei reikia naudoti kelių krypčių šešėliavimą
    - *Azimutas*: Šviesos šaltinio azimutas
    - *Aukštis*: Šviesos šaltinio aukštis

> **Pastaba**
>
> *Globalaus žemėlapio šešėliavimo* savybes galima greitai pasiekti per *Sluoksnio stiliaus* skydelį.

5.3.15 Jutiklių savybės

Kortelė *Jutikliai* naudojama konfigūruojant jutiklius ir perjungiant jų prisijungimo būseną. Kai aktyvuoti, jutikliai pasyviai rinks duomenis fone ir paskutinę duomenų versiją teiks pythono scenarijų išraiškoms.

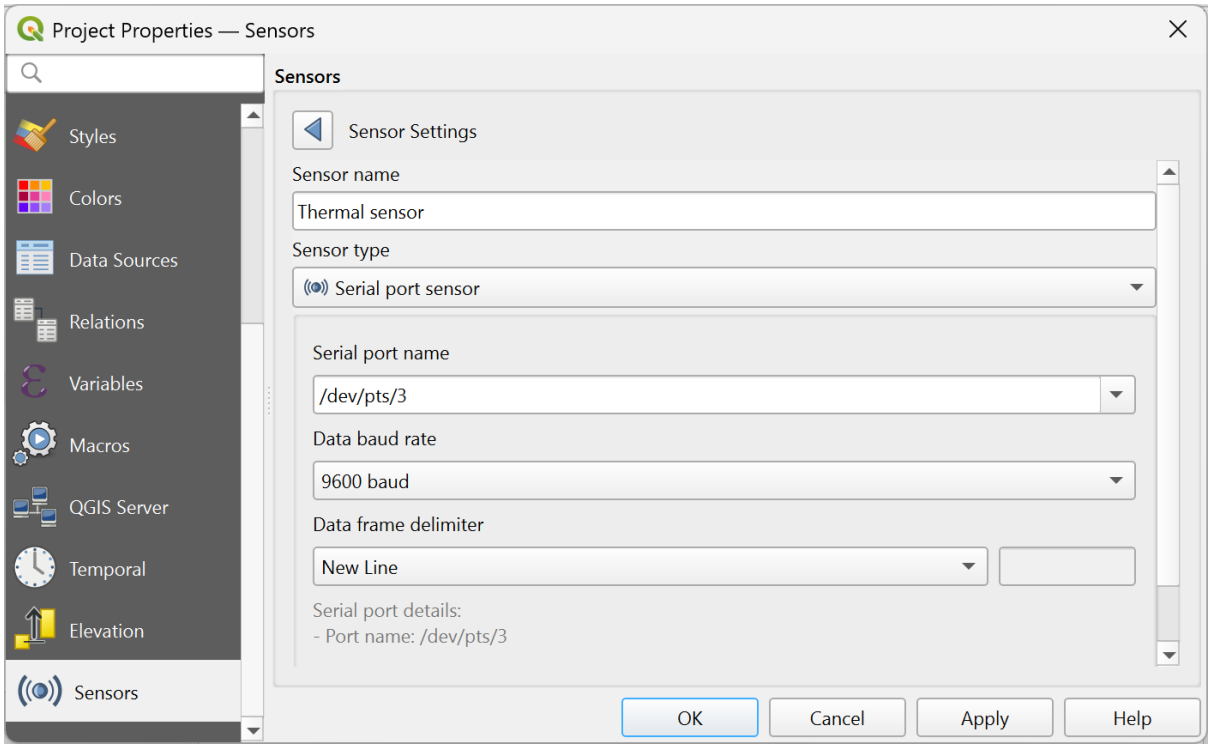


5.42 Fig. : Projekto jutiklių kortelė

Norėdami pridėti jutiklį spauskite mygtuką . Atsidarys nustatymų sub-skydelis, kuriame galėsite konfigūruoti:

- *Jutiklio pavadinimą*: kuris bus naudojamas gaunant jutiklio reikšmes python scenarijų išraiškose;
- *Jutiklio tipą*: TCP, UDP, serijinio prievado jutiklis ir pan.; ir
- papildomas nuo tipo priklausančias detales (pvz. serverio pavadinimą ir prievadą).

  Serijinio prievado jutikliams galite naudoti iškrentamuosius sąrašus su sistemoje prieinamais prievadais ir prisijungimo bodo dažniais. *Duomenų kadrų atskyriklį* galima naudoti gautų duomenų nagrinėjimui. Jei paskutiniame iš serijinio prievado išstumtame duomenų rinkinyje yra keli duomenų kadrai, grąžinami visi duomenų kadrai. Tada galite naudoti išraišką, kad toliau dalintumėte gautus paskutinius jutiklio duomenis į atskirus kadrus (pvz. naudodami funkciją *string_to_array*).



5.43 Fig. : Jutiklių nustatymų sub-skydelis

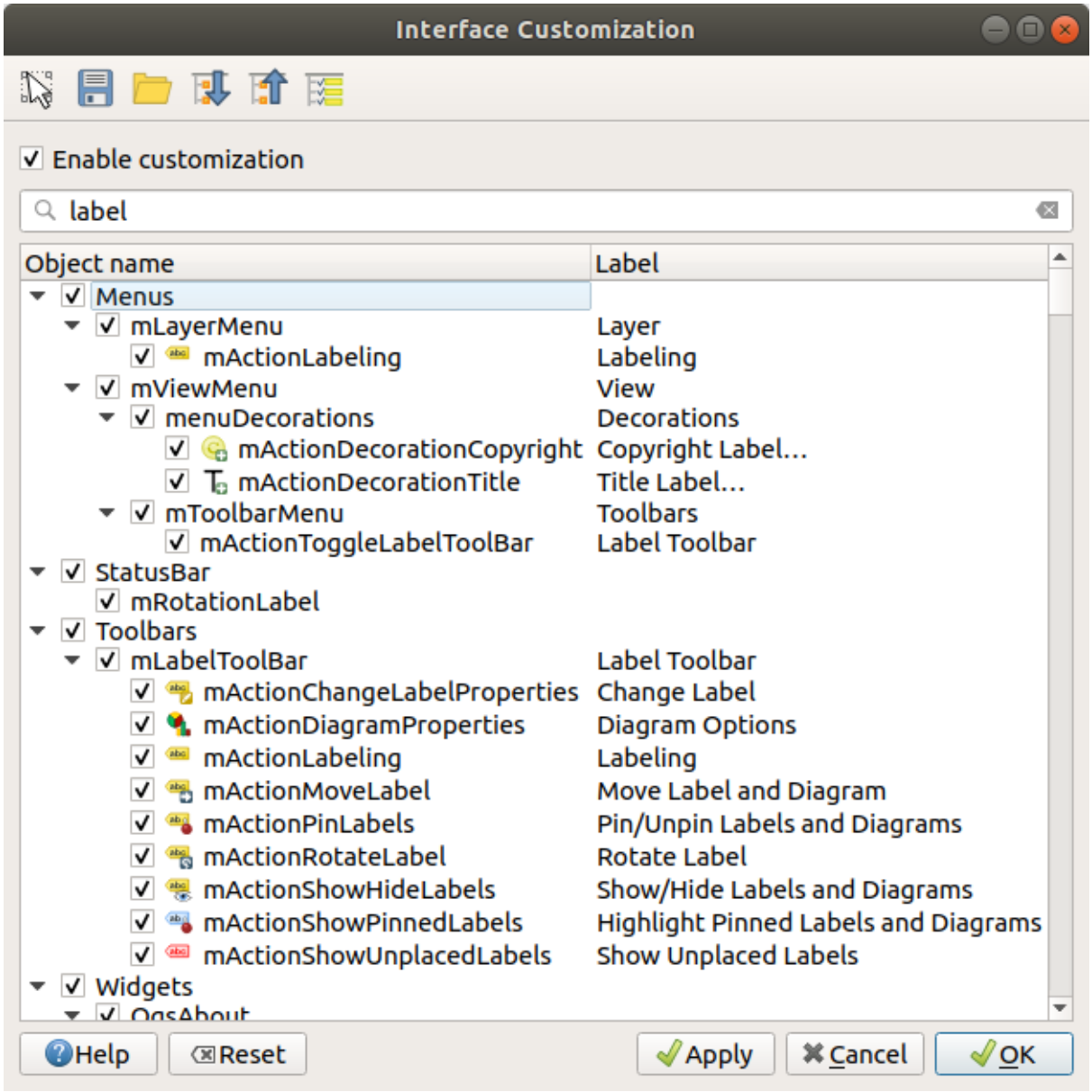
Sukonfigūravus jutiklius jūs galite naudoti mygtuką Pradėti, kad prisijungtumėte prie jutiklio. Kai jis aktyvus, paskutiniai surinkti duomenys rodomi jutiklio lentelės stulpelyje *Paskutinė reikšmė*.

5.4 Tinkinimas

Sąsajos tinkinimo dialogas leidžia (de)aktyvuoti beveik bet kurį QGIS naudotojo sąsajos elementą. Tai gali būti labai naudinga, jei norite galutiniams naudotojams pateikti „lengvesnę“ QGIS versiją, kurioje yra tik jiems reikalingos piktogramos, meniu ir skydeliai.

Pastaba

Jūsų pakeitimai bus pritaikyti tik po to, kai iš naujo paleisite QGIS.



5.44 Fig. : Naudotojo sąsajos tinkinimo dialogas

Pirmas QGIS tinkinimo žingsnis yra įjungti varnelę *Įjungti tinkinimą*. Tai įjungia įrankinę ir valdiklių skydelį, kurių pagalba jūs galite atžymėti - išjungti kai kuriuos GUI elementus.

Konfigūruojamas elementas gali būti:

- **Meniu** ar vienas iš jo submeniu pagal skyrių *Meniu juosta*
- visas **Skydelis** (žr. *Skydeliai ir įrankinės*)
- **Būsenos juosta**, aprašyta skyriuje *Būsenos juosta* arba kai kurie iš jos elementų
- **Įrankinė**: visa juosta arba kai kurios iš piktogramų
- ar bet kuris **valdiklis** iš bet kurio QGIS dialogo: užrašas, mygtukas, iškrentantis sąrašas…

Su Persijungti į elementų nurodymą pagrindinėje programoje jūs galite spausti QGIS naudotojo sąsajoje elementą, kurį norite paslėpti ir QGIS automatiškai nužymės atitinkamą įrašą Tinkinimo dialoge. Taipogi galite naudoti lauką *Ieškoti*, jei norite surasti elementus pagal jų pavadinimą ar užrašą.

Sutvarkę konfigūraciją, spauskite *Taikyti* ar *Gerai*, kad patikrintumėte jūsų pakeitimus. Ši konfigūracija bus naudojama pagal nutylėjimą, kai kitą kartą paleisite QGIS.

Pakeitimus taipogi galima įrašyti į `.ini` failą naudojant mygtuką Įrašyti į failą. Tai patogus būdas dalintis bendra QGIS sąsaja su kitais naudotojais. Tiesiog spauskite Įkelti iš failo paskirties kompiuteryje, kad importuotumėte `.ini` failą. Jūs taipogi galite vykdyti *komandinės eilutės įrankius* ir įrašyti skirtingus nustatymus skirtingiems panaudojimams.

Patarimas

Lengvai atstatykite iš anksto nustatytą QGIS

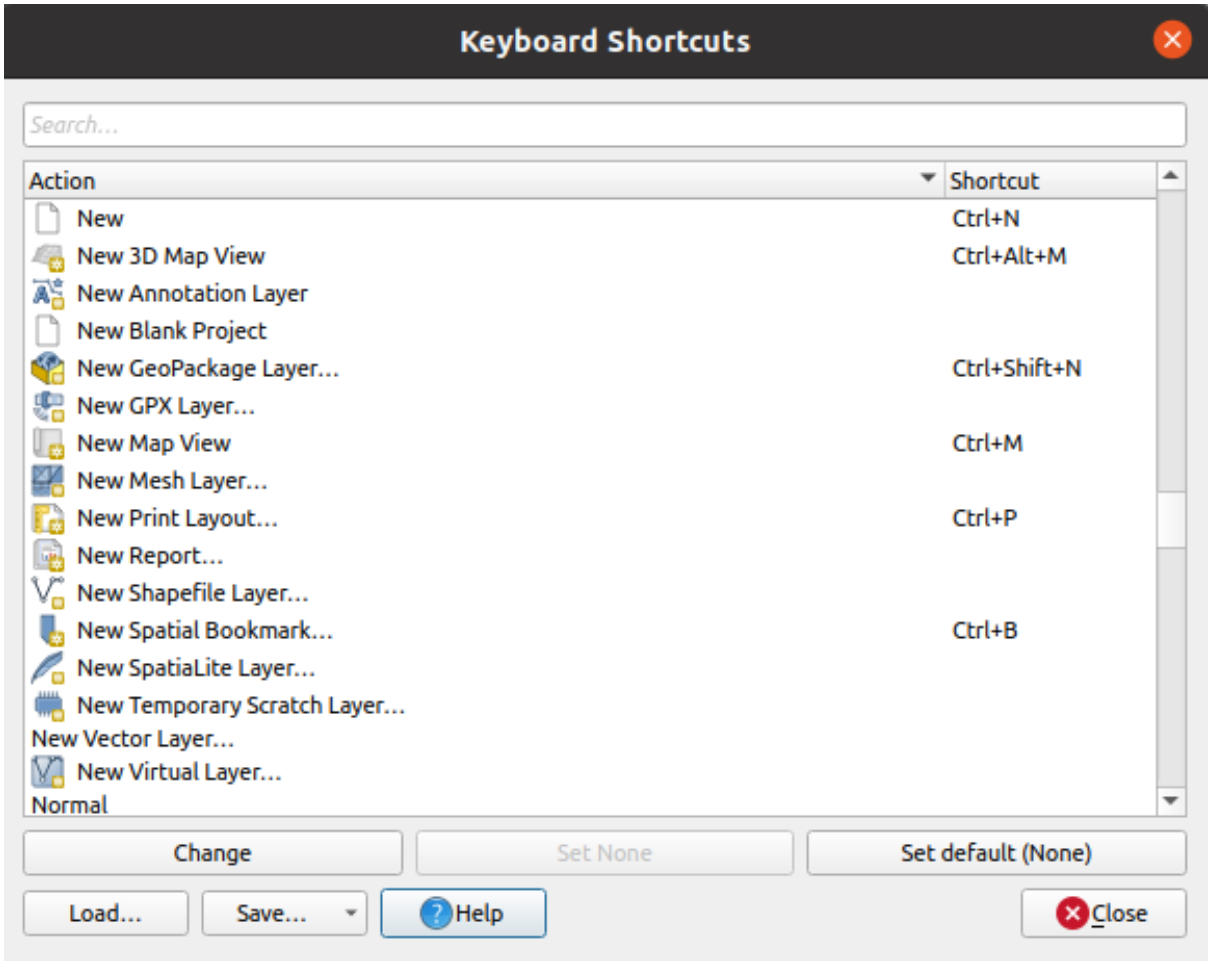
Pradinę QGIS GUI konfigūraciją galima atstatyti vienu iš žemiau pateikiamų būdų:

- tinkinimo dialoge atjunkite parinktį *Įjungti tinkinimą* arba spauskite mygtuką Parinkti viską
- spauskite mygtuką *Atstatyti* grupėje **Nustatymai**, kurią rasite *Nustatymai* ► *Parinktys* meniu dialogo kortelėje *Sistema*
- paleisti QGIS iš komandinės eilutės su šia komanda `qgis --nocustomization`
- *Nustatymai* ► *Parinktys* meniu, kortelėje *Sudėtingesni*, nustatyti kintamojo *UI* ► *Customization* ► *Enabled* reikšmę `false` (žr. *įspėjimą*).

Dažniausiai jums reikės iš naujo paleisti QGIS, kad pakeitimai būtų pritaikyti.

5.5 Klavišų kombinacijos

QGIS daugumai savybių teikia numatytąsias klavišų kombinacijas. Jas galite rasti skyriuje *Meniu juosta*. Papildomai, meniu *Nustatymai* ► *Klavišų kombinacijos…* leidžia jums pakeisti numatytąsias QGIS savybių klavišų kombinacijas ir pridėti naujas.



5.45 Fig. : Kombinacijų parinkčių apibrėžimas

Konfigūravimas yra labai paprastas. Naudokite paieškos lauką dialogo viršuje, kad rastumėte konkretų veiksmą, parinkite jį sąraše ir spauskite:

- *Keisti* ir spauskite klavišų kombinaciją, kurią norite priskirti
- *Nenustatyti jokio* jei norite išvalyti paskirtą kombinaciją
- arba *Numatytoji* jei norite grąžinti pradinę numatytąją reikšmę.

Tokiu pačiu būdu nustatykite visų kitų įrankių kombinacijas, kurias norite keisti. Baigę konfigūruoti tiesiog *Užverkite* dialogą ir jūsų pakeitimai bus pritaikyti. Taipogi galite spausti *Įrašyti* pakeitimus kaip `.XML` failą tik su Naudotojų kombinacijomis arba visomis kombinacijomis, arba kaip `.PDF` failą su visomis kombinacijomis, bei *Įkelti* jas iš kito QGIS.

5.6 QGIS paleidimas su sudėtingesniais nustatymais

5.6.1 Komandinė eilutė ir aplinkos kintamieji

Mes matėme, kad *QGIS paleidžiama* taip pat, kaip ir bet kokia kita jūsų OS aplikacija. QGIS teikia komandinės eilutės parinktis sudėtingesniems panaudos atvejams (kai kuriais atvejais jūs galite naudoti aplinkos kintamuosius vietoje komandinės eilutės parinkties). Norėdami gauti parinkčių sąrašą, komandinėje eilutėje įveskite `qgis --help`, gausite:

```
QGIS is a user friendly Open Source Geographic Information System.
Usage: /usr/bin/qgis.bin [OPTION] [FILE]
  OPTION:
        [-v, --version]     display version information and exit
        [-s, --snapshot filename]   emit snapshot of loaded datasets to given file
        [-w, --width width] width of snapshot to emit
        [-h, --height height]       height of snapshot to emit
        [-l, --lang language]       use language for interface text (changes␣
↪existing override)
        [-p, --project projectfile] load the given QGIS project
        [-e, --extent xmin,ymin,xmax,ymax]  set initial map extent
        [-n, --nologo]      hide splash screen
        [-V, --noversioncheck]      don't check for new version of QGIS at startup
        [-P, --noplugins]   don't restore plugins on startup
        [-B, --skipbadlayers]     don't prompt for missing layers
        [-C, --nocustomization]     don't apply GUI customization
        [-z, --customizationfile path]      use the given ini file as GUI␣
↪customization
        [-g, --globalsettingsfile path]     use the given ini file as Global␣
↪Settings (defaults)
        [-a, --authdbdirectory path] use the given directory for authentication␣
↪database
        [-f, --code path]   run the given python file on load
        [-F, --py-args arguments]   arguments for python. These arguments will be␣
↪available for each python execution via 'sys.argv' including the file specified␣
↪by '--code'. All arguments till '--' are passed to python and ignored by QGIS.
        [-d, --defaultui]   start by resetting user ui settings to default
        [--hide-browser]        hide the browser widget
        [--dxf-export filename.dxf]     emit dxf output of loaded datasets to␣
↪given file
        [--dxf-extent xmin,ymin,xmax,ymax]      set extent to export to dxf
        [--dxf-symbology-mode none|symbollayer|feature] symbology mode for dxf␣
↪output
        [--dxf-scale-denom scale]       scale for dxf output
        [--dxf-encoding encoding]       encoding to use for dxf output
        [--dxf-map-theme maptheme]      map theme to use for dxf output
        [--take-screenshots output_path]        take screen shots for the user␣
↪documentation
        [--screenshots-categories categories]   specify the categories of␣
↪screenshot to be used (see QgsAppScreenShots::Categories).
        [--profile name]        load a named profile from the user's profiles␣
↪folder.
        [-S, --profiles-path path]  path to store user profile folders. Will␣
↪create profiles inside a {path}\profiles folder
        [--version-migration]   force the settings migration from older version if␣
↪found
        [--openclprogramfolder]         path to the folder containing the sources␣
↪for OpenCL programs.
        [--help]                this text
        [--]            treat all following arguments as FILEs

  FILE:
    Files specified on the command line can include rasters,
    vectors, and QGIS project files (.qgs and .qgz):
     1. Rasters - supported formats include GeoTiff, DEM
        and others supported by GDAL
     2. Vectors - supported formats include ESRI Shapefiles
        and others supported by OGR and PostgreSQL layers using
        the PostGIS extension
```

Patarimas

Pavyzdys naudojant komandinės eilutės argumentus

Jūs galite paleisti QGIS nurodydami komandinėje eilutėje vieną ar daugiau duomenų failų. Pavyzdžiui jei esate aplanke `qgis_sample_data`, galite paleisti QGIS su vektorinio sluoksniu ir rastro failu, kurie bus įkelti paleidimo metu, naudodami komandą: `qgis ./raster/landcover.img ./gml/lakes.gml`

`--version`

Ši parinktis grąžina QGIS versijos informaciją.

`--snapshot`

Ši parinktis leidžia jums sukurti dabartinio vaizdo nuotrauką PNG formatu. Tai patogu, kai turite daug projektų ir norite sukurti jūsų duomenų nuotraukas arba kai jums reikia sukurti to paties projekto nuotraukas su atnaujintais duomenimis.

Šiuo metu kuriamas 800x600 taškų PNG failas. Dydį galima keisti naudojant argumentus `--width` ir `--height`. Failo pavadinimą galima pridėti po `--snapshot`. Pavyzdžiui:

```
qgis --snapshot my_image.png --width 1000 --height 600 --project my_project.qgs
```

`--width`

Ši parinktis nurodo grąžinamos nuotraukos plotį (naudojama su `--snapshot`).

`--height`

Ši parinktis nurodo grąžinamos nuotraukos aukštį (naudojama su `--snapshot`).

`--lang`

Priklausomai nuo jūsų lokalės, QGIS parenka tinkamą lokalizaciją. Jei norite pakeisti jūsų kalbą, galite nurodyti kalbos kodą. Pavyzdžiui, `qgis --lang it` paleis QGIS su Itališka lokalizacija.

`--project`

Galima paleisti QGIS su esamu projektu. Tiesiog pridėkite komandinės eilutės parinktį `--project`, po kurios nurodykite projekto pavadinimą ir tada QGIS atsidarys su visais faile nurodytais sluoksniais.

`--extent`

Naudokite šią parinktį, jei norite paleisti su konkrečia žemėlapio apimtimi. Jums reikės pridėti apimties stačiakampį tokia seka, atskiriant kableliais:

```
--extent xmin,ymin,xmax,ymax
```

Ši parinktis, tikriausiai, labiausiai tinka naudojant kartu su `--project` atidarant konkretų projektą norimoje aprėptyje.

--nologo

Ši parinktis paslepia QGIS pradžios užsklandą.

--noversioncheck

QGIS paleidimo metu netikrinti, ar yra naujesnė versija.

--noplugins

Jei paleidimo metu turite problemų su priedais, su šia parinktimi jūs galite neįkelti jų paleidimo metu. Jie vėliau vis tiek bus prieinami per Priedų tvarkyklę.

--nocustomization

Naudojant šią parinktį, nebus taikomas joks *GUI tinkinimas* QGIS paleidimo metu. Tai reiškia, kad pasirodys visi mygtukai, meniu elementai ir įrankinės, kurie yra paslėpti. Tai nėra pastovus pakeitimas. Tinkinimas vėl bus taikomas, kai QGIS bus paleistas be šios parinkties.

Parinktis naudinga laikinai leisti prieigą prie įrankių, kurie buvo išjungti tinkinimo keliu.

--skipbadlayers

Naudodami šią parinktį galite išvengti QGIS projekto paleidimo metu rodomo dialogo *Tvarkyti neprieinamus sluoksnius*. Projekto failas įkeliamas paliekant trūkstamus sluoksnius kaip neprieinamus. Daugiau informacijos apie šią temą rasite *Neteisingų kelių iki failų tvarkymas*.

--customizationfile

Naudojant šią parinktį galite nustatyti UI tinkinimo failą, kuris bus naudojamas paleidimo metu.

--globalsettingsfile

Atitinkantis aplinkos kintamasis yra `QGIS_GLOBAL_SETTINGS_FILE`.

Naudojant šią parinktį jūs galite nurodyti kelią iki Globalių nustatymų failo (`.ini`), kuris dar žinomas kaip Numatytieji nustatymai. Šiame faile esantys nustatymai pakeičia numatytuosius, bet naudotojo profilio nustatymai bus naudojami virš visko.

QGIS ieško numatytojo globalių nustatymų failo šia tvarka ir naudojamas bus tik pirmas atrastas failas:

- komandinės eilutės parametru nurodytas kelias
- aplinkos kintamuoju nurodytas kelias
- AppDataLocation aplankas, kuriame laikomi pastovūs aplikacijos duomenys; jį tvarko naudotojas arba sistemos administratorius ir jo neliečia diegimas ir jam nereikia jokio papildomo nustatymo, tokio kaip paduoti komandinės eilutės parametrus arba nustatyti aplinkos kintamąjį. Priklausomai nuo OS, tai yra:
  - `$HOME/.local/share/QGIS/QGIS3/`
  - `C:\Users\<username>\%AppData%\Roaming\QGIS\QGIS3\`
  - `$HOME/Library/Application Support/QGIS/QGIS3/`
- diegimo aplankas, pvz. `jūsų_QGIS_paketo_kelias/resources/qgis_global_settings.ini`.

Šiuo metu nėra būdo nurodyti failą, į kurį rašyti nustatymus; todėl jūs galite sukurti pradinio nustatymų failo kopiją, pervadinti jį ir pritaikyti savo reikmėms.

Nustačius `qgis_global_setting.ini` failo kelią į tinkle bendrinamą aplanką, administratoriai gali keisti globalius nustatymus ir numatytąsias parinktis keliose mašinose keičiant vieną failą.

--authdbdirectory

Ši parinktis panaši į `--globalsettingsfile`, bet nustato kelią iki aplanko, kuriame bus laikoma ir iš kurio bus įkeliama autentifikacijos duomenų bazė.

--code

Šią parinktį galima naudoti norint paleisti nurodytą python failą iš karto po QGIS paleidimo.

Pavyzdžiui jei turite python failą, pavadinimu `load_alaska.py`, su tokiu turiniu:

```
from qgis.utils import iface
raster_file = "/home/gisadmin/Documents/qgis_sample_data/raster/landcover.img"
layer_name = "Alaska"
iface.addRasterLayer(raster_file, layer_name)
```

Darant prielaidą, kad esate aplanke, kuriame yra failas `load_alaska.py`, jūs galite paleisti QGIS, įkelti rastro failą `landcover.img` ir suteikti sluoksniu pavadinimą ‚Alaska' naudodami tokią komandą:

```
qgis --code load_alaska.py
```

--py-args

Ši parinktis leidžia perduoti argumentus skriptui, pateiktam per parametrą *–code* ir bendrai kiekvienam pythono vykdymui. Po `--py-args` nebegalima perduoti jokių kitų parametrų QGIS'ui, išskyrus –. Visi argumentai, einantys po `--py-args` till `--` perduodami pythono interpretatoriui QGIS juos ignoruoja.

```
qgis --code /home/user/test.py --py-args --specialScriptArguments "a text arg"
↪'and another arg' -- layer1 layer2
```

Aukščiau pateiktame kode `test.py` turi tokį turinį `sys.argv`: [‚/home/user/test.py', ‚–specialScriptArguments', ‚tekstinis arg', ‚ir kitas arg']. `sluoksnis1` ir `sluoksnis2` bus įprastai apdorojami QGIS kaip įkeliami sluoksniai.

--defaultui

Įkėlimo metu, **visam laikui** atstato naudotojo sąsajos (UI) numatytuosius nustatymus. Ši parinktis atstatys skydelių ir įrankinių matomumą, vietą ir dydį. Jei tai vėl nebus pakeista, numatytieji UI nustatymai bus naudojami ir kitiems paleidimams.

Atkreipkite dėmesį, kad ši parinktis niekaip neįtakoja *GUI tinkinimo*. GUI tinkinimu paslėpti elementai (pvz. būsenos juosta) liks paslėpti net naudojant parinktį `--defaultui`. Taipogi žiūrėkite parinktį `--nocustomization`.

--hide-browser

Įkėlus paslepia naudotojo sąsajoje skydelį *Naršyklė*. Skydelį galima vėl įjungti spaudžiant dešinį pelės mygtuką tuščioje įrankinių vietoje arba naudojant meniu *Rodymas* ► *Skydeliai* (Linux KDE atveju *Nustatymai* ► *Skydeliai*).

Jei nebus vėl įjungtas, Naršyklės skydelis liks paslėptas ir kitų paleidimų metu.

--dxf-*

Šios parinktys gali būti naudojamos norint eksportuoti QGIS projektą į DXF failą. Galimos kelios parinktys:

- *–dxf-export*: DXF failas, į kurį eksportuojami sluoksniai;
- *–dxf-extent*: galutinio DXF failo apimtis;
- *–dxf-symbology-mode*: čia gali būti naudojamos kelios reikšmės: `none` (be simbologijos), `symbollayer` (Simbolių sluoksnio simbologija), `feature` (geoobjektų simbologija);
- *–dxf-scale-denom*: simbologijos mastelis;
- *–dxf-encoding*: failo koduotė;
- *–dxf-map-theme*: parinkite *žemėlapio temą* iš sluoksnių medžio konfigūracijos.

--take-screenshots

Padaro nuotraukas naudotojo dokumentacijai. Gali būti naudojamas kartu su `--screenshots-categories`, jei reikia filtruoti, kuriai dokumentacijos kategorijai/skilčiai reikia kurti nuotraukas (žr. QgsAppScreenShots::Categories).

--profile

Įkelia QGIS naudojant konkretų profilį iš naudotojo profilių aplanko. Jei nurodytas profilis neegzistuoja, jis bus sukurtas. Ši parinktis permuša *naudotojo pradžios profilio nustatymą*.

--profiles-path

Šia parinktimi jūs galite nurodyti kelią, kuris naudojamas įkeliant ir įrašant profilius (naudotojo nustatymus). Ji sukuria profilius aplanke `{path}\profiles`, kuriuose yra nustatymai, įdiegti priedai, apdorojimo modeliai, scenarijai ir t.t.

Ši parinktis leidžia jums, pavyzdžiui, nešti visus jūsų priedus ir nustatymus išorinėje laikmenoje arba dalintis nustatymais tarp skirtingų kompiuterių naudojant failų dalinimosi paslauga.

Atitinkantis aplinkos kintamasis yra `QGIS_CUSTOM_CONFIG_PATH`.

--version-migration

Jei randami nustatymai iš senesnės versijos (*pvz.*, `.qgis2` QGIS 2.18 aplankas), ši parinktis importuos juos į numatytąjį QGIS profilį.

--openclprogramfolder

Naudojant šią parinktį jūs galite nurodyti kitą kelią iki jūsų OpenCL programų. Tai naudinga programuotojams testuojant naujas programų versijas neperrašant esamų.

Atitinkamas aplinkos kintamasis yra `QGIS_OPENCL_PROGRAM_FOLDER`.

--

Ši parinktis leidžia perduoti tokius argumentus kaip failus, kuriuos reikia atverti QGIS'e. Tai gali būti palaikomi rastro ar vektoriniai failai, QGIS sluoksnių apibrėžimo failai ar projektų failai.

```
qgis -- /home/user/project.qgs /home/user/path_to_data/layer1.gpkg
```

Aukščiau pateiktas kodas paleis QGIS, atvers failą `project` ir įkels jame `layer1`.

5.6.2 QGIS diegimas organizacijoje

Jei jums reikia įdiegti QGIS organizacijoje su savu konfigūracijos failu, visų pirma jums reikia nukopijuoti numatytųjų nustatymų failo, esančio `jūsų_QGIS_paketo_kelias/resources/qgis_global_settings.ini`, turinį. Šiame faile jau yra kelios numatytosios skiltys, apibrėžtos blokais, prasidedančiais `[]`. Mes rekomenduojame jums išlaikyti šias numatytąsias reikšmes ir pridėti jūsų nuosavas skiltis failo apačioje. Jei sekcija faile kartojasi, QGIS panaudos paskutinę, skaičiuojant nuo viršaus apačion.

Jūs galite keisti `allowVersionCheck=false`, jei norite išjungti QGIS versijos tikrinimą.

Jei nenorite rodyti migravimo lango po šviežaus diegimo, jums reikia šios skilties:

```
[migration]
fileVersion=2
settings=true
```

Jei norite pridėti savo kintamąjį globalioje apimtyje:

```
[variables]
organisation="Your organization"
```

A lot of settings can be set using the `.INI` file such as WMS/WMTS, PostgreSQL connections, proxy settings, maptips, default values for algorithms parameters… For example, if you want to configure organization wide default values for some algorithms it would be handy if you could configure it in the prepared ini-file for all users. If the users then open the toolbox to execute an algorithm, the default values will be there. To set the default value, e.g., for `GRID_SIZE` parameter for the `native:intersection` algorithm, you could write the following into the `.INI` file:

```
[Processing]
DefaultGuiParam\native%3Aintersection\GRID_SIZE=0.01
```

Ne visus nustatymus galima paruošti per QGIS GUI, kad suprastumėte `.INI` failo nustatymų galimybes, mes siūlome jums pakeisti konfigūraciją QGIS Desktop ir tada naudodami tekstų redaktorių ieškoti jos jūsų `.INI` faile, kurį rasite jūsų profilyje.

Pabaigai jums reikia aplinkos kintamajame `QGIS_GLOBAL_SETTINGS_FILE` nurodyti kelią iki jūsų patinkinto failo.

Be to jūs galite pateikti failus, tokius kaip Python makro, spalvų paletes, išdėstymo šablonus, projekto šablonus… arba QGIS sisteminiame aplanke, arba QGIS naudotojo profilyje.

- Išdėstymo šablonai turi būti padėti į aplanką `composer_templates`.
- Projekto šablonai turi būti padėti į aplanką `project_templates`.
- Savo Python makro turi būti padėti į aplanką `python`.

SKYRIUS 6

Darbas su projekcijomis

Koordinačių atskaitos sistema, arba CRS, yra būdas susieti skaitines koordinates su Žemės paviršiaus pozicija. QGIS palaiko daugmaž 7000 standartinių CRS, kiekviena iš kurių turi skirtingą panaudojimą, privalumus ir trūkumus! Jūsų QGIS projektui tinkamos projekcijos pasirinkimas gali būti sudėtingas, laimei QGIS padeda jums šiame pasirinkime ir leidžia dirbti su skirtingais CRS kiek galima paprasčiau.

6.1 Projekcijų palaikymo apžvalga

QGIS palaiko daugmaž 7000 žinomų projekcijų. Šios standartinės CRS remiasi apibrėžtomis EPSG (European Petroleum Search Group) ir IGNF (Institut Geographique National de France). QGIS jas naudoja per projekcijų biblioteką „Proj“. Šios standartinės projekcijos paprastai identifikuojamos naudojant kombinaciją institucija:kodas, kur institucija - tai organizacijos pavadinimas, toks kaip „EPSG“ arba „IGNF“, o kodas yra unikali reikšmė, susieta su konkrečia CRS. Pavyzdžiui WGS 84 platumo/ilgumos CRS žinoma identifikatoriumi `EPSG:4326`, o žiniatinklio žemėlapių standartinė CRS yra `EPSG:3857`.

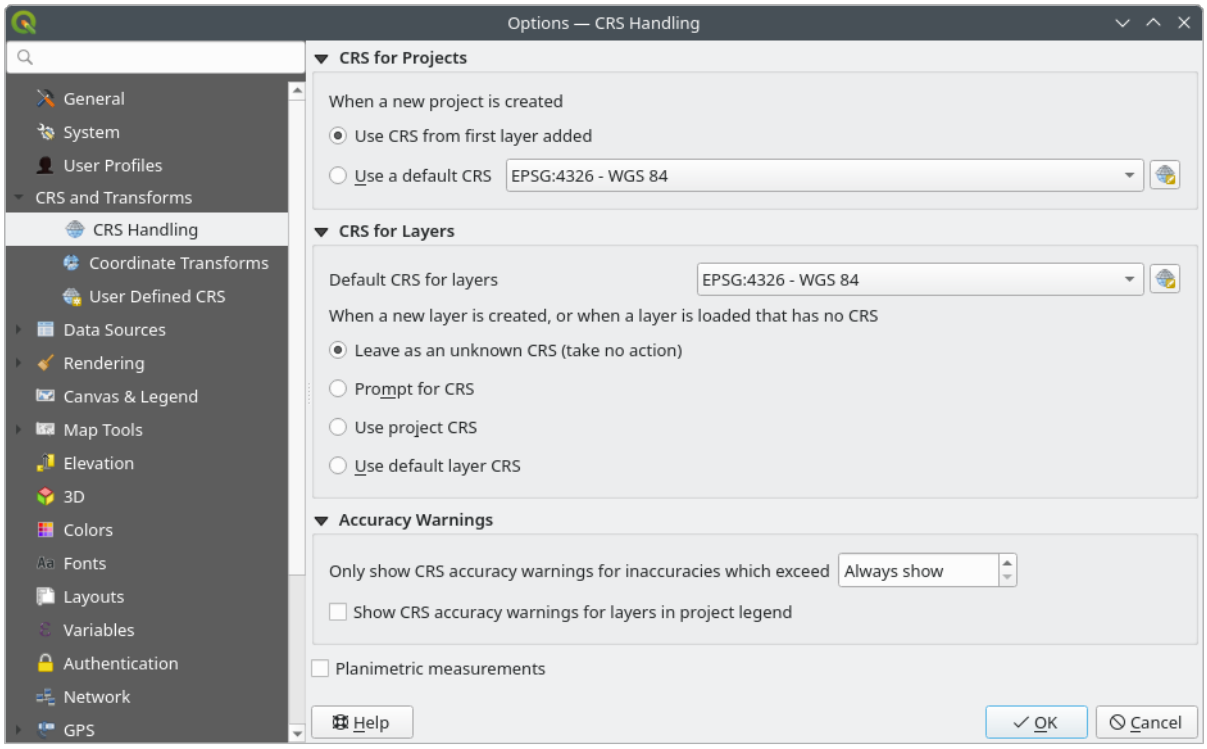
Savos, naudotojų sukurtos CRS saugomos naudotojo CRS duomenų bazėje. Skyriuje *Sava koordinačių atskaitos sistema* rasite daugiau informacijos apie jūsų koordinačių atskaitos sistemų valdymą.

6.2 Sluoksnio koordinačių atskaitos sistemos

In order to correctly project data into a specific target CRS, either your data must contain information about its coordinate reference system or you will need to manually assign the correct CRS to the layer. For PostgreSQL layers, QGIS uses the spatial reference identifier that was specified when that PostgreSQL layer was created. For data supported by GDAL, QGIS relies on the presence of a recognized means of specifying the CRS. For instance, for the Shapefile format this is a file containing an ESRI Well-Known Text (WKT) representation of the layer's CRS. This projection file has the same base name as the `.shp` file and a `.prj` extension. For example, `alaska.shp` would have a corresponding projection file named `alaska.prj`.

Kai tik sluoksnis įkeliamas į QGIS, QGIS pabando automatiškai nustatyti tinkamą sluoksnio CRS. Kai kuriais atvejais tai padaryti neįmanoma, pvz. kai sluoksnis teikiamas neišlaikant projekcijos informacijos. Jūs galite konfigūruoti, kokia turi būti QGIS elgsena, kai nepavyksta automatiškai nustatyti teisingos sluoksnio CRS:

1. Atverkite meniu *Nustatymai* ► *Parinktys…* ► *CRS valdymas*



6.1 Fig. : QGIS parinkčių dialogo CRS valdymo kortelė

2. Grupėje *Sluoksnių CRS* nustatykite veiksmą, kurį reikia atlikti *kai sukuriamas naujas sluoksnis arba kai įkeliamas sluoksnis be CRS*. Vieną iš:

   - *Palikti kaip nežinomą CRS (nieko nedaryti)*: nebus prašoma įvesti CRS, kai bus įkeliamas sluoksnis be CRS, CRS parinkimą atidedant vėlesniam laikui. Patogu, kai vienu metu įkeliama daug sluoksnių. Tokius sluoksnius *Sluoksnių* skydelyje bus galima atskirti pagal šalia jų rodomą piktogramą . Jie taipogi bus nepriršti, sluoksnio koordinatės bus traktuojamos grynai kaip skaitinės nežemiškos reikšmės, t.y. tokia pati elgsena, kaip visų sluoksnių su žyma *projektas neturi CRS*.
   - *Klausti CRS*: paprašys jūsų rankiniu būdu nurodyti CRS. Parinkti teisingą variantą yra ypač svarbu, nes neteisingas pasirinkimas sluoksnį padės blogoje Žemės paviršiaus pozicijoje! Kartais susiję metaduomenys apibrėš teisingą sluoksnio CRS, kitais atvejais turėsite susisiekti su duomenų autoriumi, kad sužinotumėte tinkamą CRS.
   - *Naudoti projekto CRS*
   - *Naudoti numatytąją sluoksnių CRS*, t.y. nustatytą viršuje esančiame iškrentančiame sąraše *Numatytoji sluoksnių CRS*.

Patarimas

Jei norite priskirti tą pačią CRS keliems sluoksniams, kuriems nepriskirta crs arba priskirta neteisinga crs:

1. Parinkite sluoksnius skydelyje *Sluoksniai*
2. Spauskite `Ctrl+Shift+C`. Jums taipogi reikia spausti dešinį pelės mygtuką ant vieno iš pažymėtų sluoksnių, kad patektumėte į *Sluoksnis* ► *Nustatyti sluoksnio(ių) CRS*
3. Raskite ir pažymėkite tinkamą naudojimui CRS
4. Spauskite *OK*. Galite patvirtinti, kad nustatyta teisingai sluoksnio savybių dialogo kortelėje *Šaltinis*.

Pastebėtina, kad CRS keitimas šiame nustatyme niekaip nekeičia šaltinio duomenų, tik pakeičia kaip QGIS interpretuoja pradines koordinates sluoksnyje dabartiniame QGIS projekte.

6.3 Projekto koordinačių ataskaitos sistemos

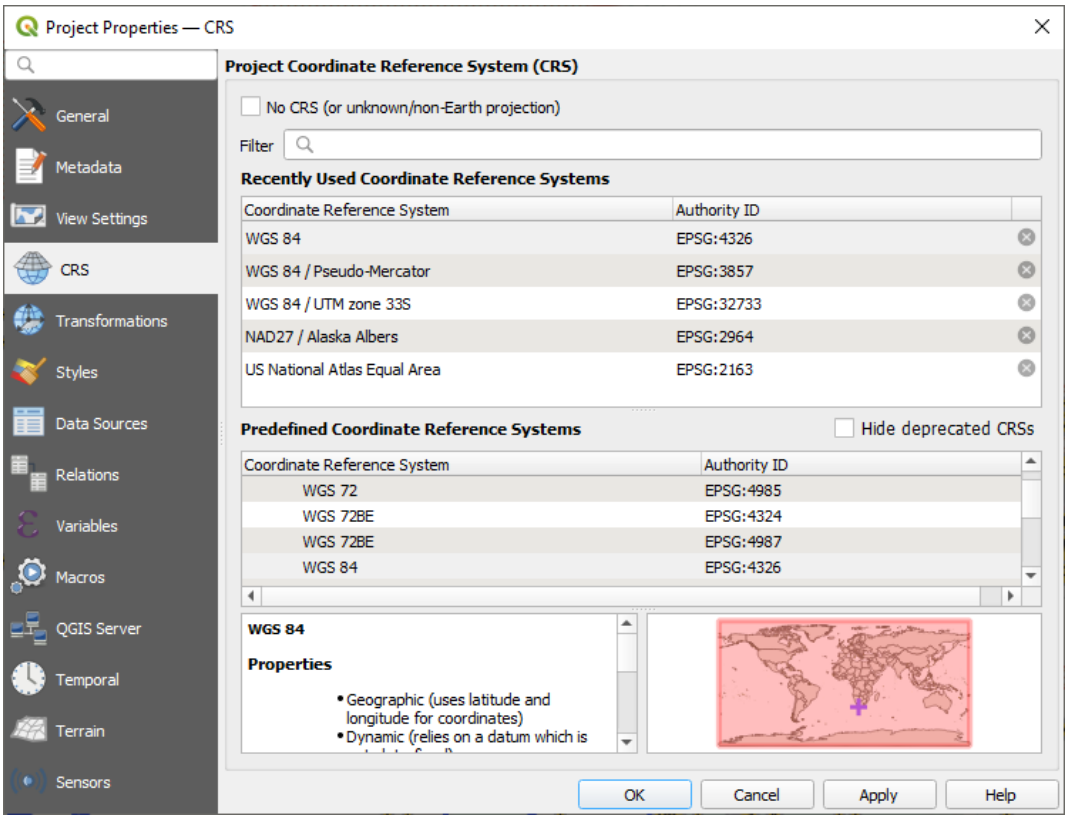
Kiekvienas QGIS projektas turi susijusią koordinačių atskaitos sistemą. Projekto CRS nustato, kaip duomenys projektuojami iš pradinių koordinačių į plokščią žemėlapį, braižomą jūsų QGIS žemėlapio drobėje.

QGIS palaiko automatinį CRS keitimą tiek rastro, tiek ir vektoriniams duomenims. Tai reiškia, kad nepriklausomai nuo konkrečių jūsų projekto žemėlapio sluoksnių CRS, jos visada bus automatiškai transformuotos į bendrą CRS, apibrėžtą jūsų projekte. QGIS fone nepastebimai perprojektuoja visus jūsų projekto sluoksnius į projekto CRS, taigi visi jie braižomi teisingoje vietoje vienas kito atžvilgiu!

Svarbu jūsų QGIS projektams parinkti tinkamą CRS. Parinkus netinkamą CRS jūsų žemėlapiai gali atrodyti iškraipyti ir prastai atspindėti tikro pasaulio objektų santykinius dydžius ir pozicijas. Paprastai dirbant su mažesniais geografiniais plotais bus naudojamos kelios standartinės CRS, pritaikytos konkrečiai šaliai ar administracinei vietai. Svarbu išsiaiškinti, kuri CRS yra tinkama ar standartinius jūsų vaizduojamos vietovės pasirinkimus, taip užtikrinant, kad jūsų QGIS projektas atitinka standartus.

Pagal nutylėjimą QGIS kiekvieną naują projektą pradeda naudojant numatytąją projekciją. Ši numatytoji CRS yra `EPSG:4326` (dar žinoma kaip „WGS 84“), tai pasaulinė platuma/ilguma paremta atskaitos sistema. Šią numatytąją CRS galima pakeisti *Naujo projekto CRS* nustatyme, kurį rasite meniu *Nustatymai* ► *Parinktys…* kortelėje *CRS valdymas* (žr. Fig. 6.1). Yra galimybė automatiškai nustatyti projekto CRS pagal pirmą į projektą įkeltą sluoksnį arba galite parinkti numatytąją CRS, kuri bus naudojama naujiems projektams. Šis pasirinkimas bus įrašytas ir naudojamas vėlesnėse QGIS sesijose.

Projekto CRS taipogi galima nustatyti per meniu *Projektas* ► *Savybės…* dialogo kortelę *CRS*. Ji taipogi bus rodoma QGIS būsenos juostos apatiniame dešiniame kampe.



6.2 Fig. : Projekto savybių dialogas

Galimos parinktys:

- *Be CRS (arba nežinoma/ne-Žemės projekcija)*: Įjungus šią parinktį, bus išjungtas VISAS QGIS projekcijų tvarkymas - visų sluoksnių koordinatės bus laikomos paprastomis 2D Dekarto koordinatėmis be jokio ryšio su jų pozicija Žemės paviršiuje. Tai gali būti naudojama spėjant sluoksnio CRS (pagal jo pradines koordinates) arba naudojant QGIS ne Žemes objektams, tokiems kaip žaidimų žemėlapiams, pastatų žemėlapiams arba mikroskopiniams dalykams. Tokiu atveju:
  - Braižant sluoksnius nevykdomas joks perprojektavimas: geoobjektai braižomi tiesiogiai naudojant jų pradines koordinates.
  - Elipsoidas išjungiamas ir priverstinai pakeičiamas į `Jokio/Planimetrinis`.
  - Atstumo ir ploto vienetai bei koordinačių rodymas išjungiami ir priverstinai rodoma „nežinomi vienetai“; visi matavimai vykdomi nežinomais žemėlapio vienetais, neleidžiamas joks konvertavimas.
- arba esama koordinačių atskaitos sistema, kuri gali būti *geografinė*, *projekcinė* arba *naudotojo apibrėžta*. Rodoma CRS apimties žemėje peržiūra, leidžianti jums pasirinkti tinkamą projekciją. Į projektą pridėti sluoksniai, nepriklausomai nuo jų pradinės CRS, automatiškai keičiami į šią CRS norint juos vaizduoti. Galima naudoti vienetus ir elipsoido nustatymus, ir jie logiški, ir jūs galite vykdyti atitinkamus skaičiavimus.

If you change your project's CRS and want the units (for distance and area) to match the units defined by the CRS, make sure you have selected *Map units* in the relevant settings, in the *General* tab of the *Project* ► *Properties…* dialog.

Patarimas

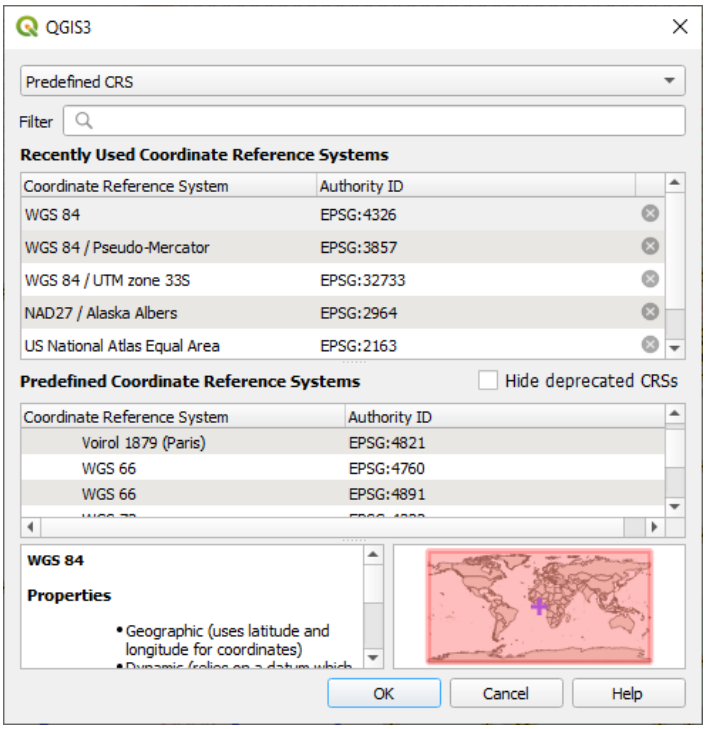
Projekto CRS nustatymas pagal sluoksnį

Jūs galite priskirti projektui CRS naudodami sluoksnio CRS:

1. Skydelyje *Sluoksniai* spauskite dešinį pelės mygtuką ant sluoksnio, iš kurio norite parinkti CRS
2. Parinkite *Nustatyti projekto CRS pagal sluoksnį*.

Projekto CRS pakeičiama į sluoksnio CRS. Pagal tai atnaujinamą žemėlapio drobės apimtis ir koordinačių rodymas. Visi projekto sluoksniai automatiškai paverčiami į naują projekto CRS.

6.4 Koordinačių atskaitos sistemos parinkiklis



6.3 Fig. : CRS parinkiklis

Šio dialogo pagalba jūs priskiriate projektui arba sluoksniui koordinačių atskaitos sistemą iš pateiktos projekcijų duomenų bazės. Dialogo elementai yra:

- **Filtras**: Jei jūs žinote koordinačių atskaitos sistemos EPSG kodą, identifikatorių ar pavadinimą, galite naudoti paieškos funkciją. Įveskite EPSG kodą, identifikatorių ar pavadinimą.
- **Paskutinės naudotos koordinačių atskaitos sistemos**: Jei turite konkrečias CRS, kurias dažnai naudojate savo kasdieniuose GIS darbuose, jos bus rodomos šiame sąraše. Spauskite ant vienos iš jų, kad parinktumėte susijusią CRS.
- **Iš anksto apibrėžtos koordinačių atskaitos sistemos**: Tai sąrašas visų QGIS palaikomų CRS, įskaitant geografines, projekcijas ir savas koordinačių atskaitos sistemas. Norėdami nustatyti CRS, parinkite ją sąraše išplėsdami atitinkamą mazgą ir parinkdami CRS. Iš pradžių būna pažymėta aktyvi CRS.
- **PROJ tekstas**: Tai yra CRS eilutė, kurią naudoja projekcijų variklis PROJ. Šis tekstas skirtas tik skaitymui ir pateikiamas tik informaciniais tikslais.

Jei norite išimti neseniai naudotą CRS iš sąrašo, tai galite padaryti keliais būdais:

- Spauskite dešinėje pusėje esantį mygtuką Išimti iš neseniai naudotų CRS sąrašo.
- Parinkę CRS, kurį norite išimti, spauskite `Del`.
- Ant naudojamo CRS spauskite dešinį pelės mygtuką ir parinkite *Išimti pažymėtą CRS iš neseniai naudotų CRS*, kad išimtumėte pažymėtą CRS arba *Valyti visus neseniai naudotus CRS*, kad išimtumėte visus CRS.

CRS parinkiklis taipogi rodo apytikslį geografinį plotą, kuriame galima naudoti parinktą CRS. Dauguma CRS yra sukurti naudojimui labai mažuose geografiniuose plotuose ir jūs neturėtumėte jų naudoti už šių plotų ribų. Peržiūros žemėlapyje užtamsinta apytikslė sąraše pažymėtos CRS naudojimo vieta. be to šiame peržiūros žemėlapyje rodomas dabartinės žemėlapio drobės apimties indikatorius.

6.5 Sava koordinačių atskaitos sistema

Jei QGIS neteikia jums reikiamos koordinačių atskaitos sistemos, jūs galite apsibrėžti savo CRS. Norėdami apibrėžti CRS, iš meniu *Nustatymai* parinkite *Savos projekcijos…*. Savos CRS saugomos jūsų QGIS naudotojo duomenų bazėje. Be jūsų savų CRS, šioje duomenų bazėje taipogi yra jūsų erdvinės žymelės ir kiti nuosavi duomenys.

Norint apibrėžti savo CRS, reikia gerai suprasti projekcijų biblioteką PROJ. Pradžiai žiūrėkite „Cartographic Projection Procedures for the UNIX Environment - A User's Manual“, parašytą Gerald I. Evenden, U.S. Geological Survey Open-File Report 90-284, 1990 (galima rasti čia https://pubs.usgs.gov/of/1990/of90-284/ofr90-284.pdf).

Šiame vadove aprašomas `proj` ir susijusių komandinės eilutės priemonių naudojimas. Kartografiniai parametrai, naudojami `proj`, aprašyti naudotojo vadove ir yra tokie patys, kaip ir naudojami QGIS.

Dialogui *Savos koordinačių atskaitos sistemos apibrėžimas* reikia tik dviejų parametrų, norint apibrėžti naudotojo CRS:

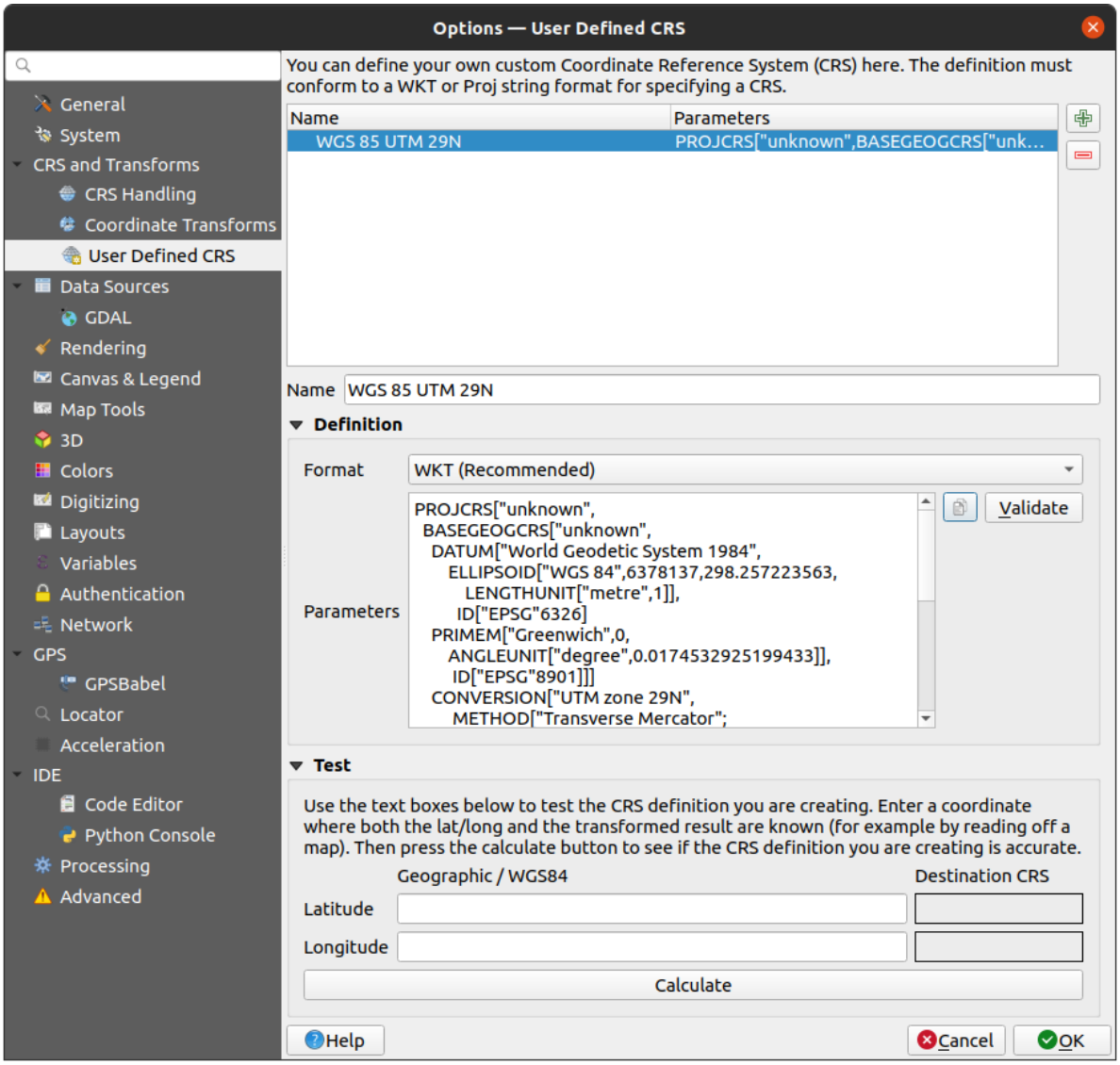
1. Apibūdinančio pavadinimo
2. Kartografinių parametrų PROJ ar WKT formatu

Kad sukurtumėte naują CRS:

1. Spauskite mygtuką Pridėti naują CRS
2. Įveskite apibūdinantį pavadinimą
3. Parinkite formatą: tai gali būti *Proj eilutė* ar *WKT*
4. Pridėkite CRS *Parametrus*.

> **Pastaba**
>
> **Pirmenybę teikite CRS apibrėžimui WKT formatu**
>
> Nors palaikomas tiek `Proj eilutės`, tiek `WKT` formatai, labai rekomenduojame projekcijos apibrėžimą saugoti WKT formatu. Todėl, jei prieinamas tik apibrėžimas proj formatu, parinkite formatą, įveskite parametrus ir tada persijunkite į WKT formatą. QGIS konvertuos apibrėžimą į WKT formatą, kurį jūs vėliau galite įrašyti.

5. Spauskite *Tikrinti*, kad patikrintumėte, ar CRS apibrėžimas yra tinkamas projekcijos apibrėžimas.



6.4 Fig. : Savos CRS dialogas

Galite patikrinti jūsų CRS parametrus ir pažiūrėti, ar jie grąžina logiškus rezultatus. Norėdami tai padaryti, įveskite žinomas WGS 84 platumos ir ilgumos reikšmes laukuose *Platuma* ir *Ilguma*. Spauskite *Skaičiuoti* ir palyginkite rezultatus su žinomomis reikšmėmis jūsų koordinačių atskaitos sistemoje.

6.5.1 NT-v2-transformacijos QGIS integracija

Norėdami integruoti NTv2 transformacijos failą į QGIS, jums reikia dar vieno žingsnio:

1. Padėkite NTv2 failą (.gsb) į CRS/Proj aplanką, kurį naudoja QGIS (pvz. `C:\OSGeo4W64\share\proj` windows naudotojams)
2. Pridėkite **nadgrids** (`+nadgrids=failopavadinimas.gsb`) į Proj apibrėžimą lauke *Parametrai*, kurį rasite *Savos koordinačių atskaitos sistemos apibrėžimas* (*Nustatymai* ► *Savo projekcijos…*).

6.5 Fig. : NTv2 transformacijos nustatymas

6.6 Duomenų transformacija

QGIS pagal nutylėjimą įjungta CRS transformacija „pagal poreikį“. Tai reiškia, kad kai jūs naudojate sluoksnius su kitomis koordinačių sistemomis, QGIS automatiškai perprojektuoja jas į projekto CRS. Kai kuriems CRS yra keli perprojektavimo į projekto CRS būdai!

Pagal nutylėjimą QGIS bandys naudoti pačią tiksliausią galimą transformaciją. Kaip ten bebūtų, kai kuriais atvejais tai gali būti neįmanoma, pvz. kai tokiai transformacijai reikia papildomų failų. Kai yra tikslesnė transformacija, bet ji šiuo metu negalima, QGIS parodys informacinį pranešimą, patariantį jums apie tikslesnę transformaciją ir kaip ją įjungti jūsų sistemoje. Paprastai tam reikia atsisiųsti išorinį transformacijų palaikymo failų paketą ir išpakuoti jį į `proj` aplanką jūsų QGIS *naudotojo profilio* aplanke.

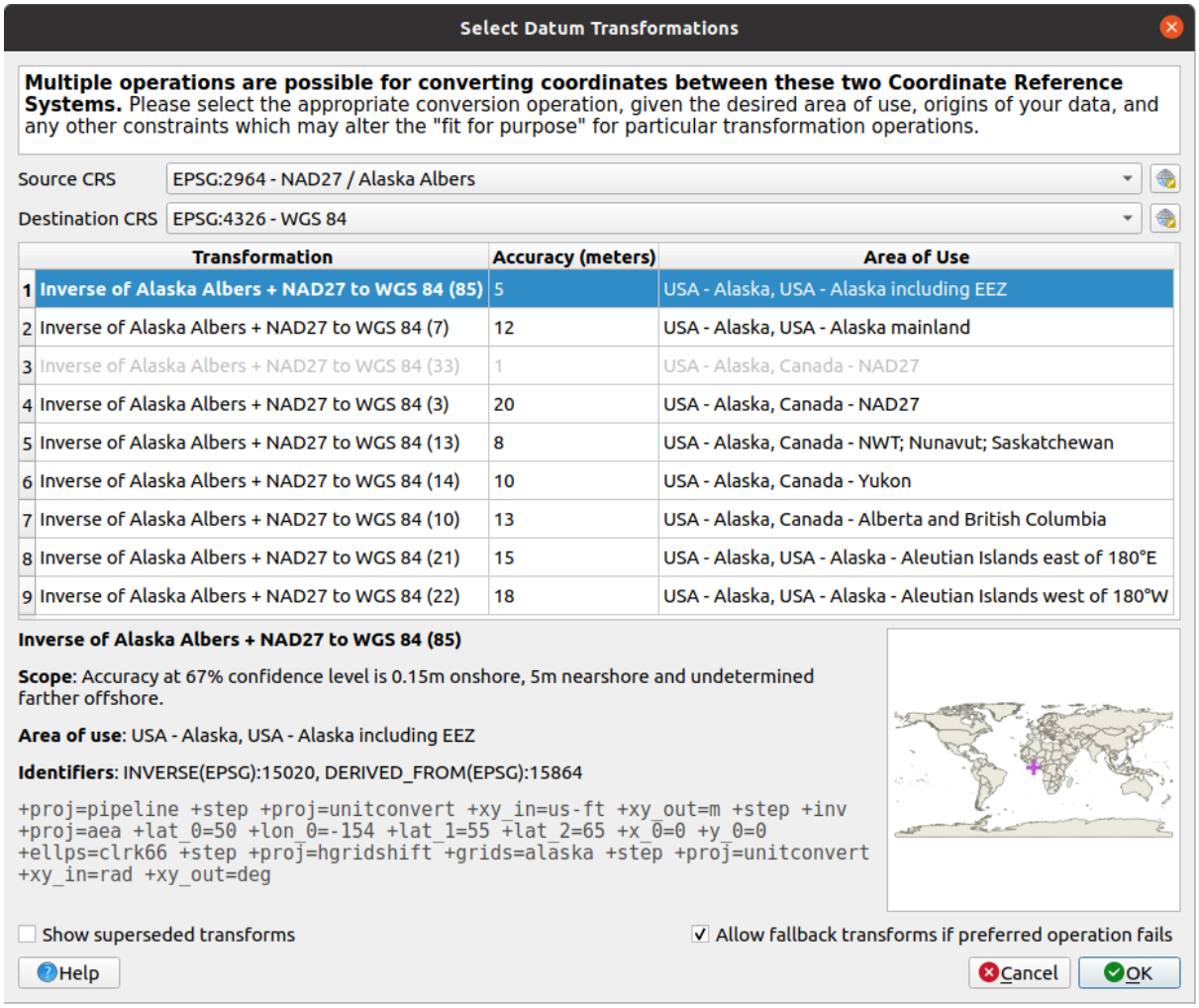
Jei reikia, QGIS taipogi gali visada pranešti, kai galimi keli transformavimo tarp dviejų CRS būdai, ir leisti jums padaryti informuotą pasirinkimą, kuri transformacija geriausiai tinka jūsų duomenims.

Šį nustatymą rasite meniu *Nustatymai* ► *Parinktys* ► *Transformacijos* kortelės grupėje *Numatytosios duomenų transformacijos*:

- naudojant parinktį *Paprašyti datum transformacijos, jei galimi keli variantai*: kai šaltinio/paskirties CRS kombinacijai yra daugiau nei vienas transformacijos variantas, automatiškai bus rodomas dialogas, kuriame bus galima pasirinkti, kurią transformaciją naudoti projekte. Jei šiame dialoge pasirenkant transformaciją įjungta parinktis *Padaryti numatytąja*, tada psirinkimas bus pritaikomas ir visiems naujai sukurtiems QGIS projektams.
- arba nustatant sąrašą tinkamų duomenų transformacijų, kurios turi būti naudojamos kaip numatytosios įkeliant sluoksnį į projektą arba perprojektuojant sluoksnį.

  Naudokite mygtuką , kad atvertumėte dialogą *Parinkti duomenų transformaciją*. Tada:

  1. Parinkite sluoksnio *Šaltinio CRS* naudojant iškrentantį meniu arba valdiklį Parinkti CRS.
  2. Tokiu pačiu būdu nurodykite ir *Paskirties CRS*.
  3. Lentelėje bus rodomas sąrašas galimų transformacijų iš šaltinio į paskirties CRS. Paspaudus eilutę bus rodomos taikomų nustatymų detalės ir atitinkantis tikslumas bei transformacijai naudojamas plotas.



6.6 Fig. : Pageidaujamos numatytosios duomenų transformacijos parinkimas

Kai kuriais atvejais transformacijos gali būti negalima naudoti jūsų sistemoje. Tokiu atveju transformacija bus rodoma šiame sąraše (pilka spalva), bet jos negalima parinkti kol neįdiegsite reikiamų paketų transformacijos palaikymui. Paprastai pateikiamas mygtukas, kurio pagalba galite atsisiųsti ir įdiegti atitinkamą tinklelį, kuris tada laikomas aktyvaus *naudotojo profilio* aplanko `proj` poaplankyje.

4. Raskite pageidaujamą transformaciją ir ją pažymėkite
5. Nurodykite, ar norite *Leisti atsargines transformacijas, jei pageidaujamos veiksmas nepavyksta*
6. Spauskite *OK*.

   Lentelėje *Numatytosios duomenų transformacijos* pridedamas įrašas su informacija apie *Šaltinio CRS*, *Paskirties CRS*, taikomu *Veiksmu* trabsformacijai ir ar įjungta *Leisti atsargines transformacijas*.

Nuo dabar QGIS automatiškai naudos parinktą duomenų transformaciją tolimesnėms transformacijoms tarp šių dviejų CRS kol jūs neišimsite jos ( ) iš sąrašo arba nepakeisite įrašo ( ).

Duomenų transformacijos nustatomos meniu *Nustatymai* ► *Parinktys* ► *Transformacijos* kortelėje, jos bus paveldimos visuose naujuose QGIS projektuose, sukurtuose jūsų sistemoje. Papildomai, konkretus projektas gali turėti savo transformacijų rinkinį, nustatytą *Projekto savybių* dialogo (*Projektas* ► *Savybės...*) kortelėje *CRS*. Šie nustatymai taikomi tik dabartiniam projektui.

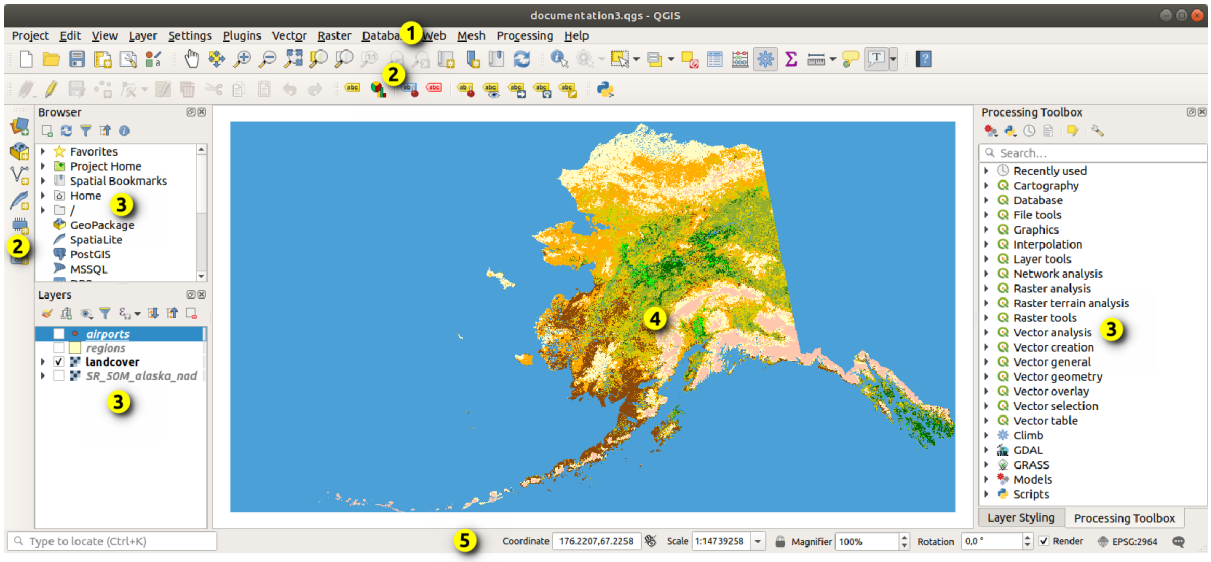
SKYRIUS 7

Žemėlapių vizualizavimas

Žemėlapių kūrimas yra vienas iš QGIS „veiklos tikslų“. Žemėlapių vaizdai yra drobės, skirtos vaizduoti į projektą įkeltus erdvinius sluoksnius pritaikius jiems įvairias braižymo parinktis. Vaizdas gali būti *2D* ar *3D* ar braižomas palei *profilio liniją*, gali rodyti skirtingus mastelius ar apimtis, rodyti skirtingą įkeltų sluoksnių rinkinį naudojant *žemėlapio temas*.

7.1 2D žemėlapio vaizdas

2D žemėlapio vaizdas (dar vadinamas **Žemėlapio drobė**) yra pagrindinė vieta, kur rodomi žemėlapiai. QGIS pagal nutylėjimą atidaromas su vienu žemėlapio vaizdu (vadinamu `pagrindiniu žemėlapiu`), kuriame sluoksniai rodomi 2D ir kuris glaudžiai susijęs su skydeliu *Sluoksniai*. Tas langas atspindi braižymą (simbologiją, užrašus, matomumus…), kurį jūs pritaikėte įkeltiems sluoksniams.



7.1 Fig. : QGIS naudotojo sąsaja

7.1.1 Žemėlapio vaizdo tyrinėjimas

Kai pridedate sluoksnį (žr. *Duomenų atidarymas*), QGIS automatiškai ieško jo CRS. Jei pagal nutylėjimą projekte nustatytas kitas CRS (žr. *Projekto koordinačių ataskaitos sistemos*), tai sluoksnio apimtis yra automatiškai verčiama į tą CRS, o žemėlapio vaizdas yra keičiamas į tą apimtį, jei pradedate tuščiu QGIS projektu. Jei projekte jau yra sluoksnių, žemėlapio drobė niekaip nekeičiama, tik pasirodys geoobjektai, patenkantys į dabartinį rodinį.

Spauskite ant žemėlapio vaizdo ir turėtumėte galėti su juo santykiauti, pastumdami į kitą plotą ar pakeisdami mastelį. guilabel:*Navigacijos įrankinėje* ir meniu *Rodymas* pateikiami reikiami įrankiai kartu su rinkiniu klavišų ar pelės mygtukų kombinacijų.

7.1 Table : Žemėlapio drobės navigacijos įrankiai

Įrankis	Naudojimas
Paslinkti žemėlapį	• Vienas kairys paspaudimas: žemėlapis centruojamas paspaustame taške tuo pačiu masteliu • Laikykite paspaudę kairį pelės mygtuką ir tempkite žemėlapio drobę.
Didinti	• Vienas kairys paspaudimas: žemėlapis centruojamas paspaustame taške, o mastelis dvigubinamas • Tempkite stačiakampį žemėlapio drobėje su kairiu pelės mygtuku, kad padidintumėte plotą. • Laikykite mygtuką `Alt`, kad persijungtumėte į įrankį Mažinti.
Mažinti	• Vienas kairys paspaudimas: žemėlapis centruojamas paspaustame taške, o mastelis mažinamas per pusę • Tempkite stačiakampį žemėlapio drobėje su kairiu pelės mygtuku, kad sumažintumėte plotą • Laikykite mygtuką `Alt`, kad persijungtumėte į įrankį Didinti.
Pastumti žemėlapį ties žymėjimu	Pastumti žemėlapį link visų skydelyje *Sluoksniai* pažymėtų sluoksnių pažymėtų geoobjektų.
Didinti iki pažymėjimo	Padidinti skydelyje *Sluoksniai* pažymėtų sluoksnių pažymėtus geoobjektus. *Taipogi prieinama per sluoksnio kontekstinį meniu*
Didinti iki sluoksnio	Didinti iki visų skydelyje *Sluoksniai* pažymėtų sluoksnių apimties. *Taipogi prieinama per sluoksnio kontekstinį meniu*
Pilnai padidinti	Didinti iki visų projekto sluoksnių apimties arba iki *pilnos projekto apimties*.
Didinti praėjusį	Didina žemėlapį iki paskutinės naudotos apimties.
Didinti kitą	Didinti žemėlapį iki kitos naudotos apimties.
Didinti iki savos rezoliucijos	Didinti žemėlapį iki tokio mastelio, kur vienas aktyvaus rastro sluoksnio pikselis dengia vieną ekrano pikselį. *Taipogi prieinama per sluoksnio kontekstinį meniu*

continues on next page

Table 7.1 – tęsinys iš praeito puslapio

Įrankis	Naudojimas
Pelės ratukas	• Stumti žemėlapį: laikykite ir tempkite pelės ratuką. • Didinti: sukite pelės ratuką kad keistumėte mastelį. Laikant paspaudus `Ctrl` kol sukamas ratukas, mastelis bus keičiamas glotniau. • Spauskite mygtukus atgal ir pirmyn, kad naršytumėte žemėlapio drobės priartinimo istoriją.
Klaviatūra	• Stumti žemėlapį: laikykite paspaudę mygtuką `Tarpas` ir judinkite pelę. Spauskite rodyklių mygtukus, kad stumtumėte žemėlapį aukštyn, žemyn, kairėn ir dešinėn. • Priartinti: Spauskite `PgUp` ar `Ctrl++` • Nutolinti: Spauskite `PgDown` ar `Ctrl+-` • Priartinti plotą: Kai aktyvūs kokie nors žemėlapio įrankiai (Identifikavimas, Matavimas…), laikykite paspaudę `Shift` ir tempkite stačiakampį ant žemėlapio, kad priartintumėte plotą. Nesuderinama su aktyviu parinkimu ar redagavimo įrankiais.

Spauskite dešinį pelės mygtuką žemėlapyje ir turėtumėte galėti *Kopijuoti koordinates* paspausto taško žemėlapio CRS, WGS84 arba savo CRS. Nukopijuota informacija vėliau gali būti įkelta į išraišką, scenarijų, tekstinį redaktorių ar skaičiuoklę…

7.1.2 Žemėlapio braižymo kontrolė

Pagal nutylėjimą QGIS braižo visus matomus sluoksnius, kai tik atnaujinama žemėlapio drobė. Įvykiai, kurie sukelia žemėlapio drobės atnaujinimą:

- sluoksnio matomumo keitimas
- matomo sluoksnio simbologijos keitimas
- sluoksnio pridėjimas
- pastūmimas ar mastelio keitimas
- QGIS lango dydžio keitimas

QGIS leidžia jums valdyti braižymo procesą keliais būdais.

- *globaliame lygyje*
- sluoksniui, naudojant pvz. *nuo mastelio priklausomą braižymą*
- arba tam skirtais GUI įrankiais.

Norėdami sustabdyti žemėlapio braižymo procesą spauskite mygtuką `Esc`. Taip bus sustabdytas žemėlapio drobės atnaujinimas ir žemėlapis bus paliktas dalinai nubraižytas. Po `Esc` paspaudimo gali užtrukti šiek tiek laiko iki braižymo sustojimo.

Norėdami sustabdyti braižymą, spauskite būsenos juostos apatiniame dešiniame kampe esančią varnelę *Braižyti*. Išjungus varnelę *Braižyti*, QGIS nebeperpaišo drobės po bet kurios iš aukščiau minėtų priežasčių. Situacijos, kada jums gali prireikti sustabdyti braižymą:

- pridėjus daug sluoksnių ir simbolizuojant juos iki braižymo
- pridėjus vieną ar daugiau didelių sluoksnių ir nustačius mastelių priklausomybę iki braižymo
- pridėjus vieną ar daugiau didelių sluoksnių ir keičiant mastelį iki reikiamo prieš braižant
- bet kokia aukščiau paminėtų situacijų kombinacija

Įjungus varnelę *Braižyti* įjungiamas braižymas bei iš karto atnaujinama žemėlapio drobė.

7.1.3 Nuo laiko priklausantis žemėlapio drobės valdymas

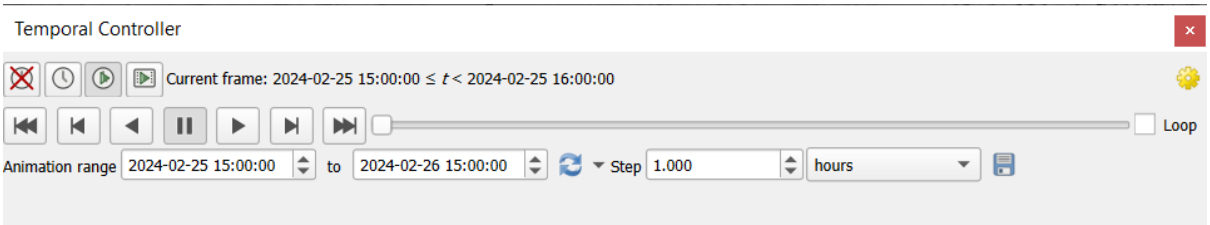
QGIS gali valdyti įkeltus sluoksnius pagal laiką, t.y. keisti žemėlapio drobę, braižant priklausomai nuo laiko kitimo. Norėdami tai pasiekti, jums reikia:

1. Sluoksnių, kurie turi nustatytas dinamines laiko savybes. QGIS palaiko laiko valdymą skirtingiems duomenų tiekėjams su savais nustatymais. Svarbiausia nustatyti laiko periodą, kuriame sluoksnyje rodoma:
   - *rastro sluoksniai*: valdo rodyti ar nerodyti sluoksnį.
   - *WMTS sluoksniai*: valdo ar duomenys turi būti braižomi pagal statinį laiko diapazoną arba seka dinaminį laiko diapazoną
   - *vektoriniai sluoksniai*: geoobjektai filtruojami priklausomai nuo susijusių laiko reikšmių atributuose
   - *tinklelio sluoksniai*: dinamiškai rodo aktyvios duomenų aibės grupės reikšmes

   Įjungus sluoksniui dinamines laiko parinktis, piktograma rodoma greta sluoksnio *Sluoksnių* skydelyje, kad primintų jums, kad sluoksnis valdomas pagal laiką. Spauskite piktogramą, jei norite keisti laiko nustatymus.
2. Įjunkite laiko navigaciją žemėlapio drobėje naudodami skydelį *Laiko valdymas*. Skydelis įjungiamas:
   - naudojant piktogramą Laiko valdymo skydelis, kurią rasite *Žemėlapio navigacijos* įrankinėje
   - arba per meniu *Nustatymai* ► *Skydeliai* ► *Laiko valdymas*
   - arba meniu *Vaizdas* ► *Duomenų filtravimas* ► *Laiko valdymas* menu

Laiko valdymo skydelis

Laiko valdymo skydelis turi šiuos režimus:



7.2 Fig. : Laiko valdymo skydelis navigacijos režime

- Išjungti laiko navigaciją: visi laiko nustatymai išjungiami ir matomi sluoksniai braižomi kaip įprasta
- Fiksuoto diapazono laiko navigacija: nustatomas laiko diapazonas ir tik sluoksniai (ar geoobjektai), kurių laiko diapazonas persidengia su šituo diapazonu yra braižomi žemėlapyje.
- Animuota laiko navigacija: nustatomas laiko diapazonas, padalintas į žingsnius, ir tik sluoksniai (ar geoobjektai), kurių laiko diapazonas persidengia su kiekvienu kadru, braižomi žemėlapyje
- Animuotas filmas: nustatomas animacijos kadrų skaičius, žemėlapio nustatymai pasikeičia su kiekvienu žingsniu, rodoma kadro po kadro eiga nedarant jokio nuo laiko priklausomo duomenų filtravimo.
- Nustatymai bendram animacijos valdymui
  - *Kadrų dažnis*: per sekundę parodomų žingsnių skaičius

- *Kaupiamasis diapazonas*: visi animacijos kadrai turės tą pačią pradžios datą ir laiką, bet skirtingas pabaigos datas ir laikus. Tai naudinga, jei norite akumuliuoti duomenis jūsų laiko vizualizacijoje, o ne rodyti per jūsų duomenis „judantį laiko langą“.

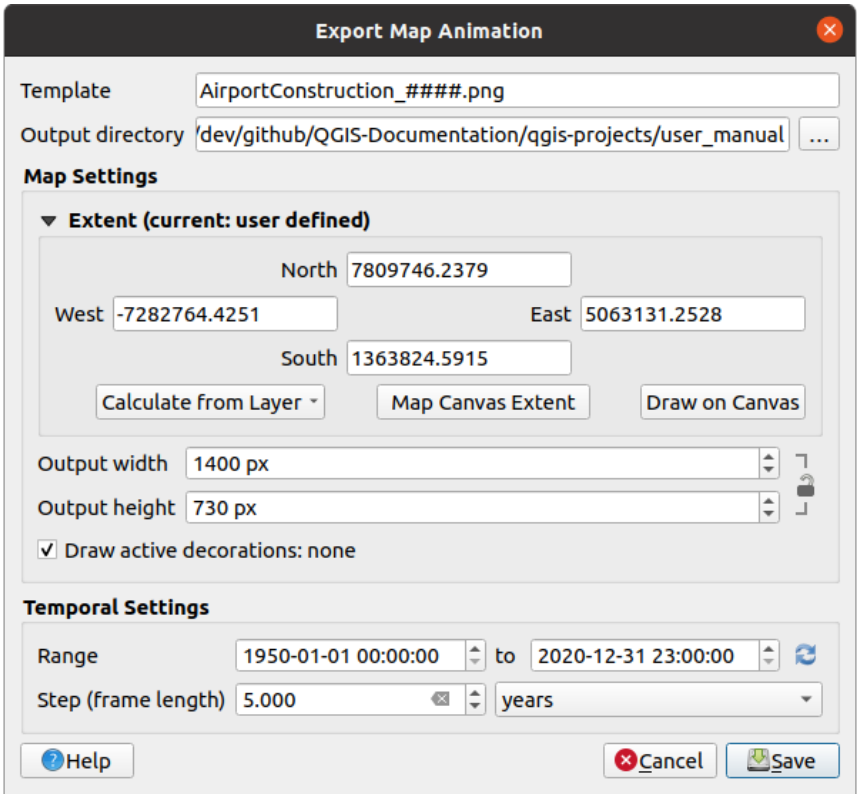
Laiko navigacijos animacija

Animacija remiasi skirtingais matomų sluoksnių rinkiniais konkrečiu laiku laiko diapazone. Norėdami sukurti laiko animaciją:

1. Įjunkite Animuota laiko navigacija, kad būtų parodytas animacijos grotuvo valdiklis
2. Įveskite reikiamą *Animacijos diapazoną*. Naudojant mygtuką , jį galima apibrėžti kaip:
   - *Nustatyti pilną diapazoną* pagal visus sluoksnius su laiku
   - *Nustatyti į išankstinį projekto diapazoną* kaip nurodyta *projekto nustatymuose*
   - *Nustatyti į vieno sluoksnio diapazoną*, paimtą iš sluoksnio su laiku
3. Įrašykite *žingsnį* laiko diapazono dalinimui. Palaikomi keli skirtingi matavimo vienetai, nuo `milisekundžių` iki `amžių`. Yra ir `šaltinio laiko` variantas, kurį parinkus laiko navigacija naudoja visus galimus projekto sluoksnio laiko diapazonus. Tai naudinga, kai projekte yra sluoksniai su laiko trūkiais, pavyzdžiui WMS-T paslauga, kuri teikia nuotraukas nepastoviomis datomis. Ši parinktis taipogi leidžia jums žengti tik per laiko diapazonus, kai rodomas kita turima nuotrauka.
4. Spauskite mygtuką norėdami peržiūrėti animaciją. QGIS sukurs scenas braižant sluoksnius nurodytais laikais. Sluoksnių rodymas priklauso nuo to, ar jie persidengia su konkrečiu laiko kadru.

   Animaciją taipogi galima peržiūrėti tempiant laiko slankiklį. Pažymėjus varnelę *Ciklas*, animacija suksis pakartotinai, kol bus paspaustas animacijos sustabdymui. Pateikiamas pilnas video grotuvo mygtukų rinkinys.

   Horizontalus slinkimas su pelės ratuku (kai palaikoma) su žymekliu ant žemėlapio drobės taipogi leis jums naviguoti arba slinkti laiko navigacijos slankiklį atgal ir pirmyn.
5. Spauskite mygtuką Eksportuoti animaciją, jei norite sukurti sceną reprezentuojančią eilę piešinių. Juos vėliau galima sujungti naudojant vaizdo redagavimo programinę įrangą:



7.3 Fig. : Žemėlapio drobės animacijos scenų eksportavimas į piešinius

- Failo pavadinimo *Šablonas*: `####` bus pakeisti kadro sekos numeriu
- *Išvesties aplankas*
- *Žemėlapio nustatymų* skiltyje jūs galite:
  - nustatyti naudojamą *erdvinę apimtį*
  - valdyti piešinių *Rezoliuciją* (*Išvesties plotį* ir *Išvesties aukštį*)
  - *Paišyti aktyvias dekoracijas*: nurodo ar aktyvios *dekoracijos* turėtų būti įtraukiamos į išvestį
- *Laiko nustatymų* skiltyje, jūs galite nurodyti:
  - animacijos *Ribas*
  - *Žingsnį (kadro ilgį)* jūsų pasirinktais vienetais

7.1.4 Aukščio valdymo skydelis

Aukščio valdymas leidžia jums tvarkyti aukščio z-diapazono duomenis 2D žemėlapiuose. Ši parinktis šiuo metu palaiko taškų masyvo sluoksnius ir rastro DEM. Naudokite skydelį, jei jums reikia vaizduoti ir sąveikauti su aukščio duomenimis žemėlapio drobėje.

Įjunkite *Aukščio valdymo* skydelį žemėlapio drobėje paspausdami *Duomenų filtravimas*, kurį rasite *Rodymo* meniu. *Aukščio valdymą* matysite kaip diapazono slankiklį kairėje žemėlapio drobės pusėje. Slankiklio viršuje yra Nustatymų meniu, kur jūs galite:

- Nustatyti *Fiksuotą diapazono dydžio* reikšmę, kad užrakintumėte aukščio diapazoną ties konkrečiu dydžiu. Šis nustatymas įrašomas į projektą ir išlaikomas tarp skirtingų sesijų, užtikrinant jūsų aukščio filtro pastovumą. Naudokite parinktį *Apsukti kryptį*, kad apsuktumėte aukščio filtro slankiklį, kad jis judėtų nuo didesnių reikšmių apačioje iki žemesnių reikšmių viršuje.

- *Nustatyti aukščio diapazoną*, kad atvertumėte projekto aukščio kortelę, kur galėsite nustatyti aukščio diapazono apatinę ir viršutinę ribas
- *Išjungti aukščio filtrą*, kad išjungtumėte aukščio filtrą iš žemėlapio drobės

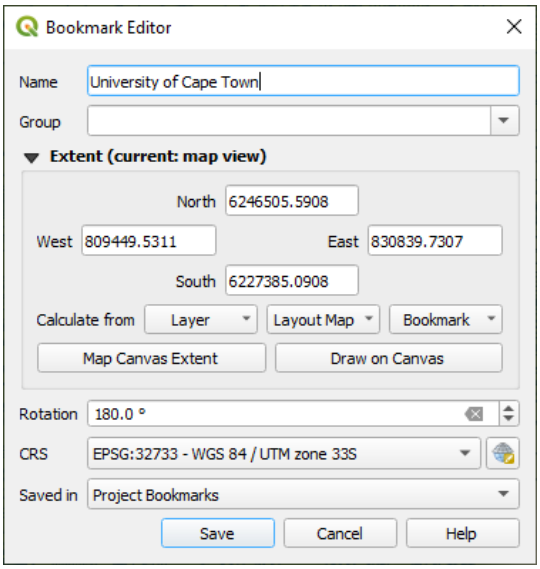
7.1.5 Žemėlapio apimčių žymelės

Erdvinės žymelės leidžia jums „pasižymėti“ geografinę vietą ir grįžti į ją vėliau. Pagal nutylėjimą žymelės įrašomos į naudotojo profilį (kaip *Naudotojo žymelės*), tai reiškia, kad jos matomos visuose naudotojo atvertuose projektuose. Jos taipogi gali būti įrašytos į konkretų projektą (kaip *Projekto žymelės*) ir laikomos projekto faile, tai gali būti naudinga, jei projektu dalinamasi su kitais naudotojais.

Žymelės sukūrimas

Norėdami sukurti žymelę:

1. Pritraukite norimą vietą.
2. Parinkite meniu *Vaizdas* ► *Nauja erdvinė žymelė…*, spauskite `Ctrl+B` arba dešiniu mygtuku paspauskite *Erdvinių žymelių* įrašą *Naršyklės* skydelyje ir parinkite *Nauja erdvinė žymelė*. Atidaromas dialogas *Žymelių redaktorius*.



7.4 Fig. : Žymelių redagavimo dialogas

3. Įveskite žymelę aprašantį pavadinimą
4. Įveskite ar parinkite pavadinimą grupės, kurioje reikia laikyti susijusias žymeles
5. Parinkite apimtį vietos, kurią norite įrašyti, naudodami valdiklį *apimties parinkiklis* widget
6. Pakeiskite žemėlapio *Pasukimą*
7. Nurodykite apimčiai naudojamą *CRS*
8. Parinkite, ar žymelę *Įrašyti* į *Naudotojo žymeles* ar *Projekto žymeles* (pagal nutylėjimą šio iškrentančio sąrašo reikšmė yra *Naudotojo žymelės*)
9. Spauskite *Įrašyti*, kad pridėtumėte žymelę į sąrašą

Pastebėtina, kad galite turėti kelias žymeles tokiu pačiu pavadinimu.

Darbas su žymelėmis

To use and manage bookmarks, you can either use the *Spatial Bookmarks* panel or *Browser*:

- Select *View* ► *Show Spatial Bookmark Manager* or press `Ctrl+7` to open the *Spatial Bookmark Manager* panel.
- Select *View* ► *Show Spatial Bookmarks* or press `Ctrl+Shift+B` to show the *Spatial Bookmarks* entry in the *Browser* panel.

Jūs galite vykdyti tokias užduotis:

7.2 Table : Žymelių valdymo veiksmai

Užduotis	Erdvinių žymelių tvarkyklė	Naršyklė
Priartinti žymelę	Spauskite du kartus ant žymelės arba parinkite žymelę ir spauskite mygtuką *Didinti ties žymele*.	Spauskite du kartus ant žymelės arba tempkite ir numeskite ją į žemėlapio drobę, arba spauskite dešiniu mygtuku ant žymelės ir parinkite *Didinti ties žymele*.
Trinti žymelę	Parinkite žymelę ir spauskite mygtuką *Trinti erdvinę žymelę*. Patvirtinkite jūsų pasirinkimą.	Spauskite dešinį mygtuką ant žymelės ir parinkite *Trinti erdvinę žymelę*. Patvirtinkite jūsų pasirinkimą.
Eksportuoti žymeles į XML	Spauskite mygtuką *Importuoti/Eksportuoti žymeles* ir parinkite *Eksportuoti*. Visos žymelės (naudotojo ar projekto) bus įrašytos į xml failą.	Parinkite vieną ar daugiau aplankų (naudotojo ar projekto) arba poaplankius (grupes), tada spauskite dešiniu mygtuku ir parinkite *Eksportuoti erdvines žymeles…*. Įrašomas parinktas žymelių poaibis.
Importuoti žymeles iš XML	Spauskite mygtuką *Importuoti/Eksportuoti žymeles* ir parinkite *Importuoti*. Visos žymelės XML faile importuojamos kaip naudotojo žymelės.	Spauskite dešinį mygtuką ant *Erdvinės žymelės* arba vieno iš jo aplankų (naudotojo ar projekto) arba poaplankių (grupių), kad nurodytumėte, kur reikia importuoti žymeles. Tada parinkite *Importuoti erdvins žymeles*. Jei vykdoma įrašui *Erdvinės žymelės*, žymelės pridedamos į *Naudotojo žymeles*.
Keisti žymelę	Jūs galite pakeisti žymelę, keisdami reikšmes lentelėje. Jūs galite pakeisti pavadinimą, grupę, apimtį ir ar ji laikoma projekte, ar ne.	Spauskite dešinį mygtuką ant norimos žymelės ir parinkite *Keisti erdvinę žymelę…*. Bus atidarytas *Žymeklių redaktorius*, kuriame jūs galėsite pakeisti bet kurį žymelės aspektą taip, tarsi kurtumėte naują žymelę. Jūs taipogi galite tempti žymelę iš vieno į kitą aplanką (naudotojo ar projekto) ar poaplankį (grupę).

Jūs galite valdyti žymelių veiksmus paspaudę dešinį mygtuką ant norimos *Erdvinių žymelių tvarkyklės* žymelės. Jūs taipogi galite priartinti žymeles įrašę žymelės pavadinimą į *lokatorių*.

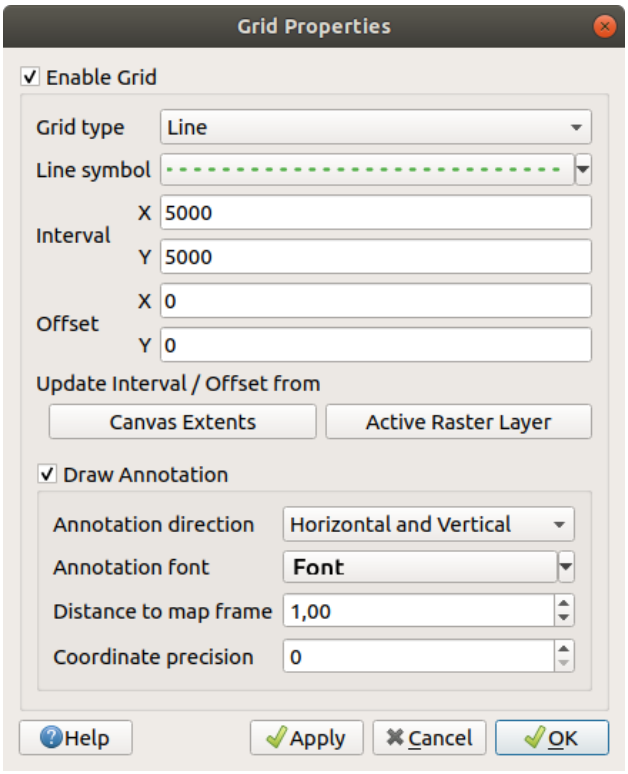
7.1.6 Žemėlapio dekoravimas

Dekoracijomis vadinamas tinklelis, antraštė, autorių teisės, piešiniai, šiaurės rodyklė, mastelio juosta ir išdėstymo apimtys. Jie naudojami žemėlapio „dekoravimui“ pridedant kartografinius elementus.

Tinklelis

Tinklelis leidžia jums pridėti į žemėlapio drobę koordinačių tinklelį ir koordinačių anotacijas.

1. Parinkite meniu *Rodymas* ► *Dekoracijos* ► *Tinklelis…*, kad atvertumėte dialogą.



7.5 Fig. : Tinklelio dialogas

2. Pažymėkite *Įjungti tinklelį* ir nurodykite tinklelio parametrus pagal sluoksnius, įkeltus į žemėlapio drobę:
   - *Tinklelio tipas*: gali būti *Linija* arba *Žymeklis*
   - Susijęs *Linijos simbois* arba *žymeklio simbolis* naudojamas tinklelio žymėms
   - *Intervalas X* ir *Intervalas Y* tarp tinklelio žymų žemėlapio vienetais
   - *Poslinkis X* ir *Poslinkis Y* - tinklelio žymų atstumai nuo apatinio kairiojo žemėlapio drobės kampo, žemėlapio vienetais
   - Intervalo ir poslinkio parametrus galima nurodyti priklausomus nuo:
     - *Drobės apimties*: kuria tinklelį su intervalais, kurie yra daugmaž 1/5 drobės pločio
     - *Aktyvaus rastro sluoksnio* rezoliucijos
3. Įjunkite *Piešti anotaciją*, jei norite ant tinklelio rodyti koordinates, nurodykite:
   - *Anotacijos kryptį*, t.y. kaip reikia dėti užrašus tinklelio atžvilgiu. Tai gali būti:
     - *Horizontaliai* arba *Vertikaliai* visiems užrašams

  - *Horizontaliai ir vertikaliai*, t.y. kiekvienas užrašas yra lygiagretus su juo susijusiai tinklelio žymai
  - *Ribos kryptis*, t.y. kiekvienas užrašas seka drobės ribą ir yra statmenas savo tinklelio žymai
- *Anotacijos šriftas* (teksto formatavimas, buferis, šešėlis…) naudojant valdiklį *šrifto parinkimas*
- *Atstumas iki žemėlapio rėmo*, paraštė tarp anotacijų ir žemėlapio drobės krašto. Patogu, kai *žemėlapio drobė eksportuojama* pvz. į piešinio formatą ar PDF, siekiant išvengti anotacijų ant „popieriaus“ ribų.
- *Koordinačių tikslumas*

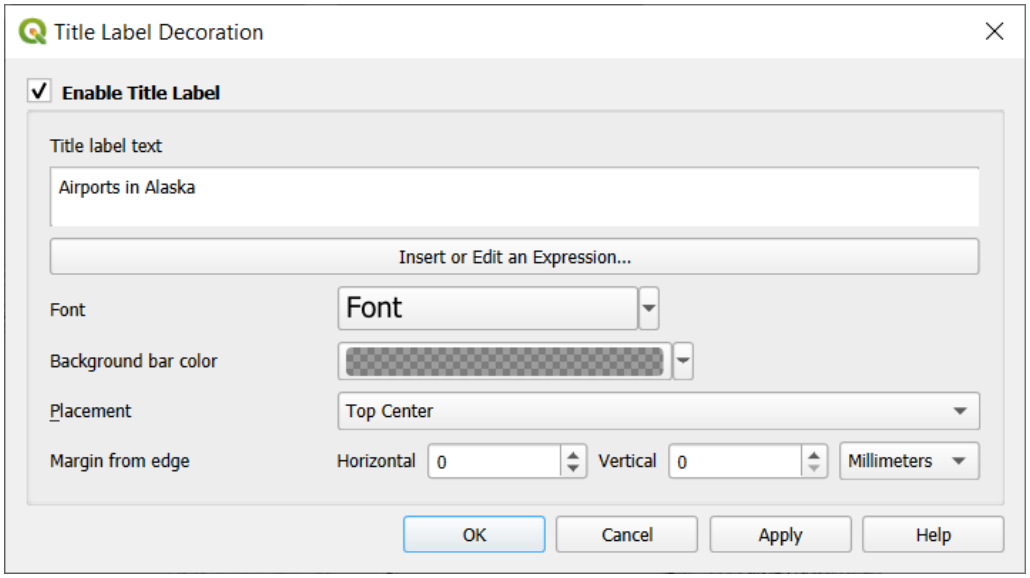
4. Spauskite *Taikyti*, kad patikrintumėte, ar atrodo taip, kaip norėjote arba *Gerai*, jei jums viskas tinka.

Antraštės užrašas

Antraštės užrašas leidžia jums dekoruoti jūsų žemėlapį su **Antrašte**.

Norėdami pridėti Antraštės užrašo dekoraciją:

1. Parinkite meniu *Rodymas* ► *Dekoracijos* ► *Antraštės užrašas…*, kad atvertumėte dialogą.



7.6 Fig. : Antraštės dekoracijos dialogas

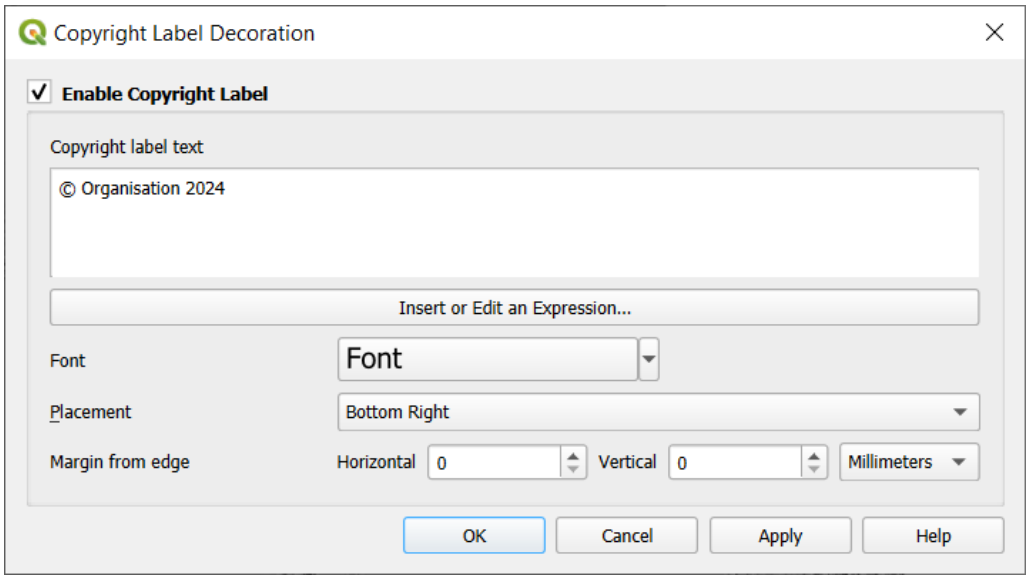
2. Įsitikinkite, kad įjungta varnelė *Įjungti antraštės užrašą*
3. Įveskite antraštės tekstą, kurį norite įdėti į žemėlapį. Galite padaryti jį dinamišku, naudodami mygtuką *Įterpti ar keisti išraišką…*.
4. Parinkite užrašo *Šriftą* naudodami *šrifto parinkimo valdiklį* su pilna prieiga prie QGIS *teksto formatavimo* parinkčių. Greitai nustatykite šrifto spalvą, permatomumą spausdami juodą rodyklę šrifto iškrentančio sąrašo dešinėje.
5. Parinkite *spalvą*, kuri bus naudojama antraštės *Fono juostai*.
6. Parinkite užrašo *Vietą* drobėje: galimi variantai yra *Viršuje kairėje*, *Viršuje centre* (numatytoji parinktis), *Viršuje dešinėje*, *Apačioje kairėje*, *Apačioje centre* ir *Apačioje dešinėje*.
7. Patikslinkite elemento vietą parinkdami horizontalią ir/arba vertikalią *Paraštę nuo krašto*. Jas galima nurodyti **Milimetrais** arba **Taškais (pikseliais)** ar nurodyti kaip **Procentus** žemėlapio drobės pločio ar aukščio.
8. Spauskite *Taikyti*, kad patikrintumėte, ar atrodo taip, kaip norėjote arba *Gerai*, jei jums viskas tinka.

Autorių teisių užrašas

Autorių teisių užrašą galima naudoti jūsų žemėlapio dekoravimui su **Autorių teisių** užrašu.

Norėdami pridėti šią dekoraciją:

1. Parinkite meniu *Rodymas* ► *Dekoracijos* ► *Autorių teisių užrašas…*, kad atvertumėte dialogą.



7.7 Fig. : Autorių teisių dekoracijos dialogas

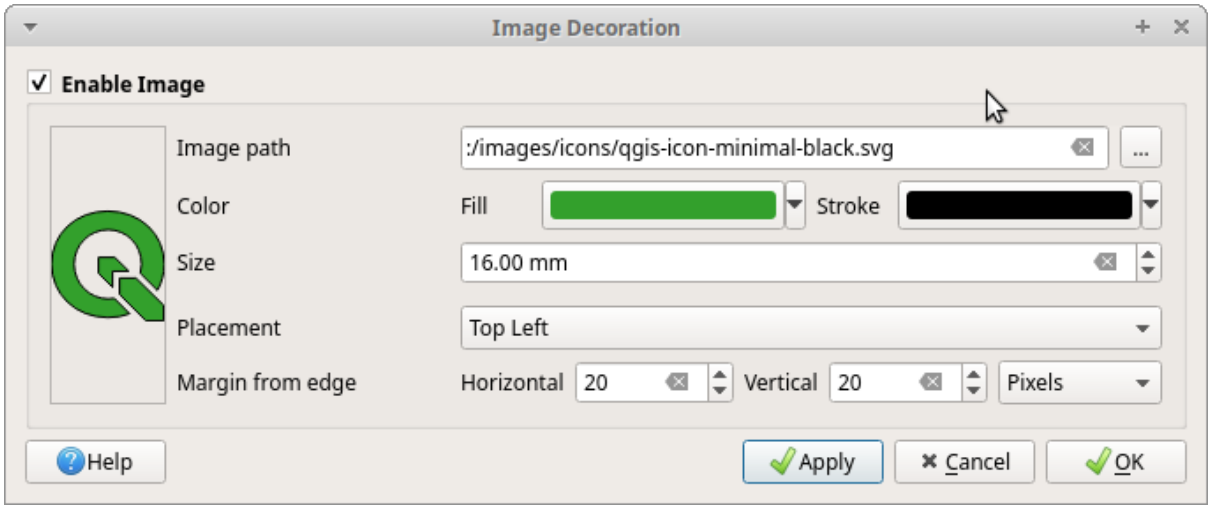
2. Įsitikinkite, kad įjungta varnelė *Įjungti autorių teisių užrašą*
3. Įveskite autorių teisių tekstą, kurį norite įdėti į žemėlapį. Galite padaryti jį dinamišku, naudodami mygtuką *Įterpti ar keisti išraišką…*.
4. Parinkite užrašo *Šriftą* naudodami *šrifto parinkimo valdiklį* su pilna prieiga prie QGIS *teksto formatavimo* parinkčių. Greitai nustatykite šrifto spalvą, permatomumą spausdami juodą rodyklę šrifto iškrentančio sąrašo dešinėje.
5. Parinkite užrašo *Vietą* drobėje: galimi variantai yra *Viršuje kairėje*, *Viršuje centre*, *Viršuje dešinėje*, *Apačioje kairėje*, *Apačioje centre*, and *Apačioje dešinėje* (numatyta autorių teisių užrašo parinktis)
6. Patikslinkite elemento vietą parinkdami horizontalią ir/arba vertikalią *Paraštę nuo krašto*. Jas galima nurodyti **Milimetrais** arba **Taškais (pikseliais)** ar nurodyti kaip **Procentus** žemėlapio drobės pločio ar aukščio.
7. Spauskite *Taikyti*, kad patikrintumėte, ar atrodo taip, kaip norėjote arba *Gerai*, jei jums viskas tinka.

Piešinio dekoracija

Piešinys leidžia jums žemėlapio drobėje pridėti piešinį (logotipą, legendą, ..).

Norėdami pridėti piešinį:

1. Parinkite meniu *Rodymas* ► *Dekoracijos* ► *Piešinys…*, kad atvertumėte dialogą.



7.8 Fig. : Piešinio dekoracijos dialogas

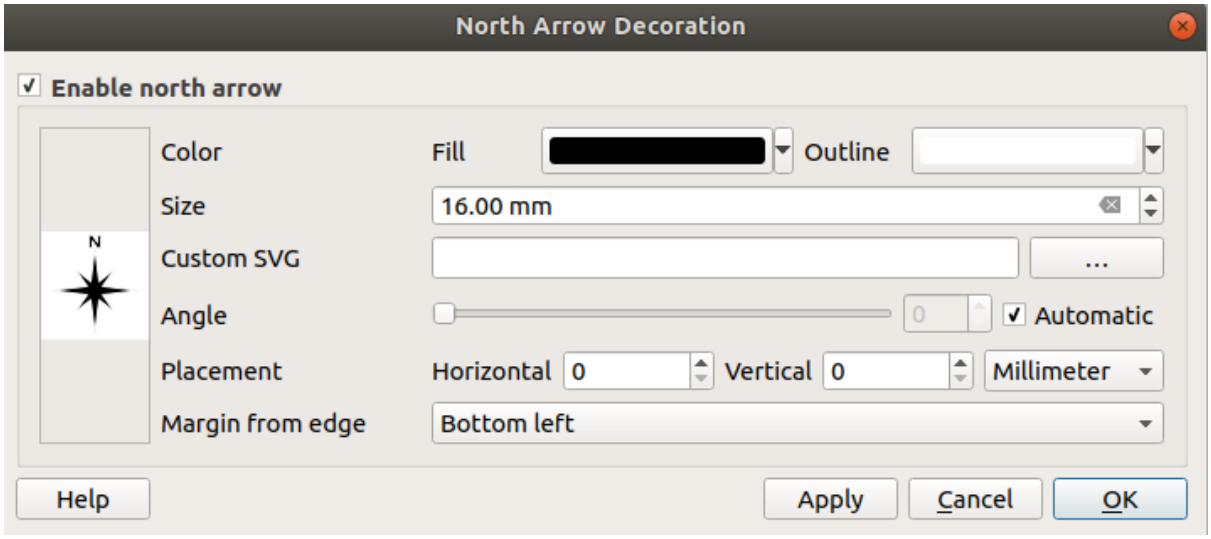
2. Įsitikinkite, kad įjungta varnelė *Įjungti piešinį*
3. Parinkite rastro (pvz. png ar jpg) ar SVG piešinį, naudojant mygtuką … Naršyti
4. Jei parinkote nuo parametrų priklausomą SVG, tada galite taipogi nustatyti *Užpildymo* ar *Brėžimo* (kontūro) spalvą. Rastro piešiniams spalvų nustatymai išjungti.
5. Nustatykite piešinio *Dydį* milimetrais. Parinkto piešinio plotis naudojamas keičiant piešinio dydį į nurodytą *Dydį*.
6. Parinkite, kur norite padėti piešinį žemėlapio drobėje, naudodami iškrentantį sąrašą *Vieta*. Numatytoji vieta yra *Viršuje kairėje*.
7. Nustatykite the *Horizontalią* ir *Vertikalią* paraštes nuo (drobės) krašto. Šias reikšmes galima nurodyti **Milimetrais**, **Taškais (pikseliais)** arba nurodyti **Procentus** drobės pločio ar aukščio.
8. Spauskite *Taikyti*, kad patikrintumėte, ar atrodo taip, kaip norėjote ir *Gerai*, jei jums viskas tinka.

Šiaurės rodyklė

Šiaurės rodyklė leidžia jums pridėti šiaurės rodyklę į žemėlapio drobę.

Kad pridėtumėte šiaurės rodyklę:

1. Parinkite meniu *Rodymas* ► *Dekoracijos* ► *Šiaurės rodyklė…*, kad atvertumėte dialogą.



7.9 Fig. : Šiaurės rodyklės dialogas

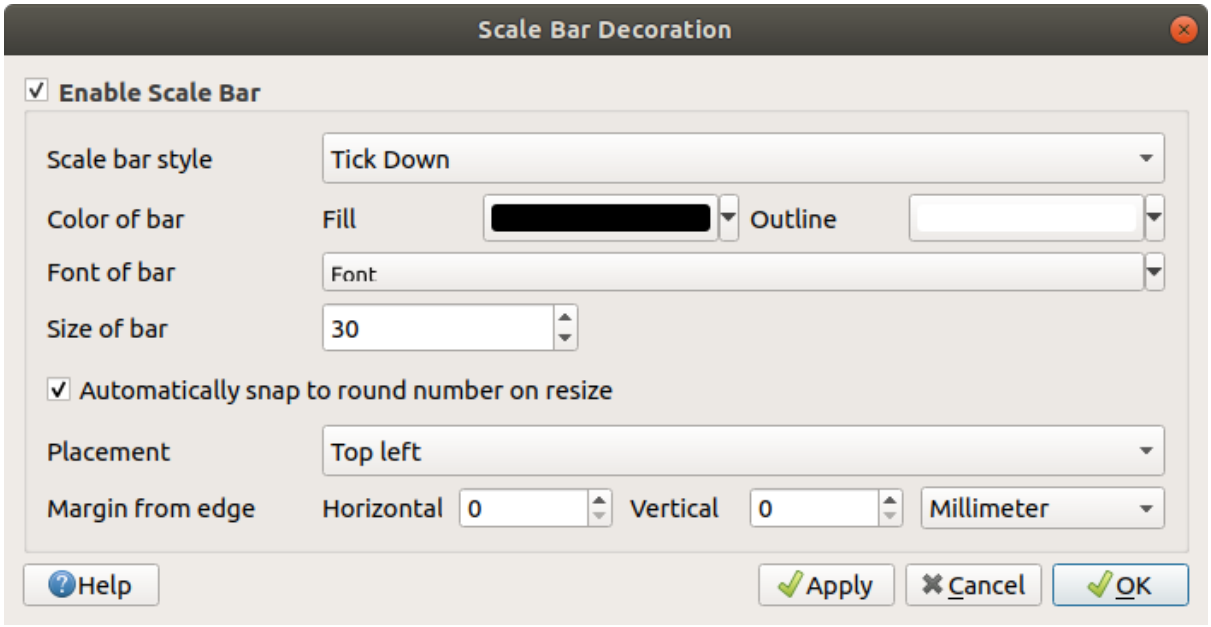
2. Įsitikinkite, kad įjungta varnelė *Įjungti Šiaurės rodyklę*
3. Pasirinktinai pakeiskite spalvą ir dydį arba parinkite savo SVG
4. Pasirinktinai pakeiskite kampą ar pasirinkite **Automatinis**, kad QGIS nustatytų kryptį
5. Pasirinktinai pasirinkite vietos iškrentančiame sąraše pasirinkite vietą
6. Pasirinktinai patikslinkite rodyklės vietą nurodydami horizontalią ir/ar vertikalią *Paraštę nuo (drobės) krašto*. Šias reikšmes galima nurodyti **Milimetrais** ar **Taškais (pikseliais)** arba nurodyti **Procentą** nuo drobės pločio ar aukščio.
7. Spauskite *Taikyti*, kad patikrintumėte, ar atrodo taip, kaip norėjote ir *Gerai*, jei jums viskas tinka.

Mastelio juosta

Mastelio juosta prideda į žemėlapio drobę paprastą mastelio juostą. Jūs galite valdyti juostos stilių, vietą bei užrašus. Mastelio juosta atsižvelgia į aktyvius *projekto atstumo vienetus*, kaip nurodyta *Projekto savybėse* ► *Bendra* ► *Atstumo matavimo vienetai*.

Norėdami pridėti mastelio juostą:

1. Parinkite meniu *Rodymas* ► *Dekoracijos* ► *Mastelio juosta…*, kad atvertumėte dialogą



7.10 Fig. : Mastelio juostos dialogas

2. Įsitikinkite, kad įjungta varnelė *Įjungti mastelio juostą*
3. Iškrentančiame sąraše parinkite *Mastelio stilių*
4. Parinkite *Juostos spalvą*, parinkdami užpildymo spalvą (pagal nutylėjimą juoda) ir rėmo spalvą (pagal nutylėjimą balta). Mastelio juostos užpildymą ir rėmą galima padaryti permatomą, spaudžiant ant rodyklės žemyn spalvos parinkimo valdiklyje.
5. Mastelio juostos šriftą parinkite iškrentančiame sąraše *Juostos šriftas*
6. Nustatykite *Juostos dydį* aktyviais vienetais
7. Pasirinktinai įjunkite *Automatiškai pritraukti prie apvalaus skaičiaus keičiant dydį*, jei norite gauti lengviau skaitomas reikšmes
8. Iškrentančiame sąraše *Vieta* parinkite juostos vietą
9. Galite patikslinti elemento vietą nurodydami horizontalią ir/ar vertikalią *Paraštė nuo (drobės) krašto*. Šias reikšmes galima nurodyti **Milimetrais** ar **Taškais (pikseliais)**, arba galite nurodyti žemėlapio drobės pločio ar aukščio **Procentus**.
10. Spauskite *Taikyti*, kad patikrintumėte, ar atrodo taip, kaip norėjote arba *Gerai*, jei jums viskas tinka.

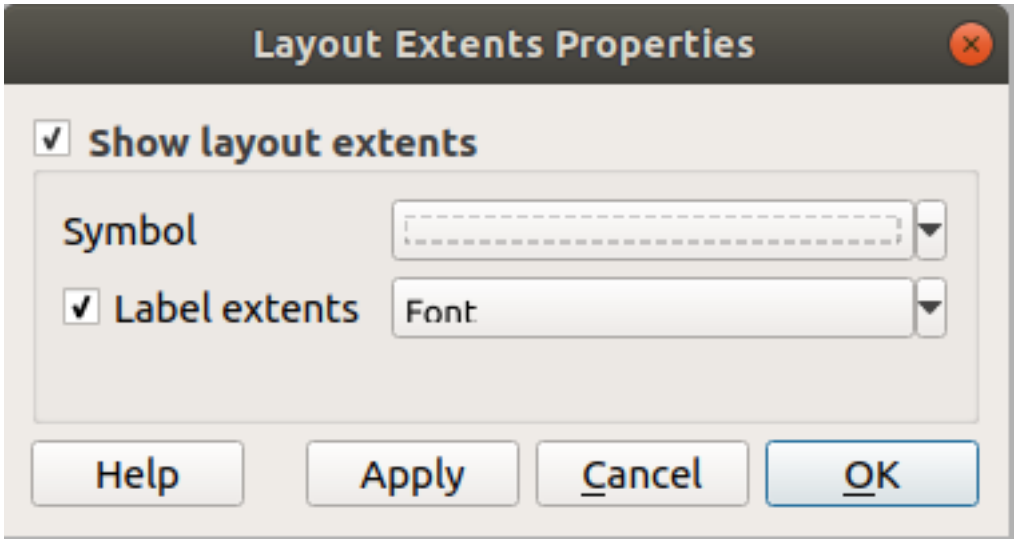
Išdėstymų apimtys

Išdėstymo apimtys prideda į drobę spausdinimo maketo *žemėlapio elementų* apimtis. Įjungus šią parinktį, visų spausdinimo maketų, visų išdėstymų žemėlapių elementų apimtys bus rodomi lengvai taškuotu kontūru su spausdinimo maketo žemėlapio elemento pavadinimu. Galite valdyti rodomų išdėstymo elementų užrašų stilių. Ši dekoracija naudinga, kai tvarkote pozicijas žemėlapio elementų, tokių kaip užrašai, kai jums reikia suprasti realias spausdinimo išdėstymuose matomų elementų apimtis.

7.11 Fig. : Išdėstymo apimčių, rodomų QGIS projekte su dviem spausdinimo išdėstymais, pavyzdys. Spausdinimo išdėstymas vardu „Vaizdai“ turi du žemėlapio elementus, o kitame spausdinimo išdėstyme yra tik vienas žemėlapio elementas.

Norėdami pridėti išdėstymo apimtį(is):

1. Parinkite *Rodymas* ► *Dekoracijos* ► *Išdėstymų apimtys*, kad atvertumėte dialogą



7.12 Fig. : Išdėstymo apimčių dialogas

2. Įsitikinkite, kad įjungta varnelė *Rodyti išdėstymo apimtis*.

3. Pasirinktinai galite keisti apimčių simbolį ir užrašus.
4. Spauskite *Taikyti*, kad patikrintumėte, ar atrodo taip, kaip norėjote ir *Gerai*, jei jums viskas tinka.

Patarimas

Dekoracijų nustatymai

Kai įrašote QGIS projekto failą, bet kokie jūsų padaryti Tinklelio, Šiaurės rodyklės, Mastelio juostos, Autorių teisių ar Išdėstymų apimčių pakeitimai bus įrašyti į projektą ir atstatyti kitą kartą įkėlus projektą.

7.1.7 Anotacijų įrankiai

Anotacijos - tai dar vienas tipas elementų, kuriuos galima pridėti į žemėlapio drobę norint suteikti papildomą informaciją, kurios negali atvaizduoti braižomi sluoksniai. Priešingai nei *užrašai*, kuriems reikia vektoriniuose sluoksniuose esančių atributų reikšmių, anotacijos yra nepriklausomos detalės, saugomos pačiame projekto faile.

QGIS turi dvi anotacijų šeimas:

- **Geoobjektų anotacijos**: tai tikri teksto, žymeklių, linijų ar poligonų geoobjektai su geografinėmis pozicijomis, saugomi specialiame sluoksnyje, pavadintame „anotacijų sluoksnis“. Jie pririšti prie konkrečios geografinės vietos, tai reiškia, kad paslinkus jūsų žemėlapį, pakeitus mastelį ar projekciją, anotacijos nešokinės po jūsų žemėlapį. Jos visada bus fiksuotos konkrečioje vietoje, kurioje jas nubraižėte.
- **Balionų anotacijos**: tai individualios HTML ar formos anotacijos burbulo viduje. Jas galima susieti su bet kuriuo sluoksniu matomumui ir braižymui ant žemėlapio drobės. Dydis priklauso nuo žemėlapio drobės mastelio ir jo poziciją galima prikabinti.

Patarimas

Žemėlapio išdėstymas su anotacijomis

Jūs galite spausdinti ar eksportuoti anotacijas kartu su jūsų žemėlapiu į skirtingus formatus naudojant:

- žemėlapio drobės eksportavimo įrankius, kuriuos rasite meniu *Projektas*
- *spausdinimo išdėstymą*, kur jums gali tekti atitinkamo žemėlapio elemento savybėse įjungti parinktį *Braižyti žemėlapio drobės elementus*

Anotacijų įrankinė teikia aibę įrankių, leidžiančių kurti ir valdyti abi anotacijų šeimas.

7.3 Table : Anotacijų įrankinės veiksmai

Įrankis	Naudojimas	Sritis
Naujas anotacijų sluoksnis	Kurti naują sluoksnį anotacijų saugojimui	*Feature annotations*
Pagrindinio anotacijų sluoksnio savybės	Valdo pagrindinio anotacijų sluoksnio savybes	
Keisti anotacijas	Parinkti, perkelti, keisti dydį ar keisti anotacijų simbolizavimo savybes	
Kurti poligono anotaciją	Kurti anotaciją kaip poligono geoobjektą	
Kurti linijos anotaciją	Kurti anotaciją kaip poli-linijos geoobjektą	
Kurti žymeklio anotaciją	Kurti anotaciją kaip taško geoobjektą	
Kurti teksto anotaciją taške	Create an annotation as a text label anchored at a point	
Kurti tekstinę anotaciją palei liniją	Kurti anotaciją kaip palei linijos geoobjektą išlenktą tekstą	

continues on next page

Table 7.3 – tęsinys iš praeito puslapio

Įrankis	Naudojimas	Sritis
Create Text Annotation in Rectangle	Create an annotation as a text label in a rectangular text box	*Balloon annotations*
Kurti paveikslo anotaciją	Create an annotation representing an `.SVG` or raster image	
HTML anotacija	Select and create annotation with a `HTML` formatted text	
Formos anotacija	Pažymėti ir kurti anotaciją, rodančią savo formos failo vektorinio sluoksnio atributus	

Pastaba

Nuo QGIS 3.40 nebegalima kurti SVG ar teksto tipų baliono anotacijas. Įkeliant senus projektus, bet kurios tokios anotacijos bus automatiškai keičiamos į naujesnius piešinių ir stačiakampio teksto anotacijų tipus.

Geoobjektų anotacijos

Geoobjektų anotacijos saugomos **anotacijų sluoksniuose**. Priešingai nei standartiniai sluoksniai, anotacijų sluoksnis prieinamas tik dabartiniame projekte ir jame gali būti skirtingų tipų geoobjektai (tekstas, žymekliai, linijos, poligonai, piešiniai). Sluoksnis neturi atributų ir susijusios simbologijos, vietoje to kiekvienas geoobjektas gali būti simbolizuojamas atskirai per skydelį *Sluoksnio stilius*.

QGIS turi du anotacijų sluoksnių tipus:

- Bendras *Anotacijų sluoksnis*: jūs galite jį sukurti su įrankiu Naujas anotacijų sluoksnis. Sukurtą sluoksnį rasite *Sluoksnių* skydelyje, kur galėsite valdyti jo geoobjektų matomumą, perkelti jo matomumą virš ar po kitų jūsų žemėlapio sluoksnių, kaip bet kuriam kitam sluoksniui. Du kartus paspaudus ant sluoksnio pasieksite jo savybes.
- *Pagrindinis anotacijų sluoksnis*: Pagal nutylėjimą anotacijos laikomos šiame sluoksnyje, kai projekte nėra anotacijų sluoksnio anotacijos sukūrimo metu nepažymėtas nei vienas anotacijų sluoksnis. Šis sluoksnis visada rodomas pačiame žemėlapio viršuje ir jūs nepamatysite jo skydelyje *Sluoksniai* su kitais projekto sluoksniais, tai reiškia, kad jo geoobjektai visada matomi. *Anotacijų* įrankinės mygtukas *Pagrindinio anotacijų sluoksnio savybės* padeda jums atverti jo savybių dialogą.

Sluoksnio savybės

Anotacijų sluoksnio savybių dialoge yra šios kortelės:

- *Informacija*: tik skaitymui skirtas dialogas, kuriame yra įdomi vieta greitai peržvelgti dabartinio sluoksnio apibendrintą informaciją ir metaduomenis. Čia gali būti sluoksnio apimtis, bendras ir skirtingų anotacijų tipų elementų skaičius, CRS detalės, …
- *Šaltinis*: apibrėžia bendrus anotacijos sluoksnio nustatymus. Jūs galite:
  - Nustatyti *Sluoksnio pavadinimą* kuris bus naudojamas sluoksnio identifikavimui projekte (*Sluoksnių skydelyje*, išraiškose, …)
  - Rodyti sluoksnio *Priskirtą koordinačių atskaitos sistemą (CRS)*: jūs galite pakeisti sluoksnio CRS, pažymėdami neseniai naudotą iškrentantčiame sąraše arba paspaudę mygtuką Parinkti CRS (žr. *Koordinačių atskaitos sistemos parinkiklis*). Naudokite šį procesą tik jei sluoksniui taikoma CRS yra netinkama arba nėra nurodyta apskritai.
- *Braižymas*:

- Galite nurodyti *Maksimalų (imtinai)* ir *Minimalų (išskirtinai)* mastelį, taip nurodydami, kuriuose masteliuose bus rodomi geoobjektai. Nepatekę į šį diapazoną objektai paslepiami. Mygtukas Nustatyti pagal dabartinį drobės mastelį padeda jums panaudoti dabartinį žemėlapio drobės mastelį kaip matomumo ribą. Daugiau informacijos rasite *Nuo mastelio priklausomo matomumo parinkimas*.
- *Susijęs sluoksnis*: leidžia jums pasirinktinai nurodyti su anotaciją susijusį matomumo sluoksnį. Jį nurodžius, anotacijos bus braižomos tik tada, kai žemėlapyje matomas susietas sluoksnis.
- *Nepermatomumas*: Naudodami šį įrankį jūs galite žemiau esantį sluoksnį matomu. Naudokite slankiklį, kad pritaikytumėte jūsų vektorinio sluoksnio matomumą pagal poreikius. Jūs taipogi galite tiksliai nurodyti matomumo procentą laukelyje greta slankiklio.
- *Suliejimo režimas Sluoksnio* lygyje: Naudodami šiuos įrankius jūs galite pasiekti specialius braižymo efektus, kuriuos anksčiau žinojote tik grafikos programose. Aukščiau ir žemiau esančių sluoksnių taškai maišomi pagal nustatymus, aprašytus *Suliejimo režimai*.
- Pritaikyti *paišymo efektus* visiems sluoksnio geoobjektams mygtuku *Paišyti efektus*.

Kai kurios iš šių parinkčių prieinamos iš geoobjektų anotacijų *Simbologijos* savybių.

Sąveika su geoobjektais

Kiekvienas anotacijų tipas turi specialiai jam skirtą sukūrimo įrankį.

7.4 Table : Geoobjekto anotacijos kūrimas

Šeima / paremtas	Anotacijos įrankis	Procedūra
Geometrija	Kurti poligono anotaciją	Tai kaip taško, linijos ar poligono vektorinio geoobjekto vektorizavimas. Kuriant anotacijų elementus veikia visos įprastos QGIS klavišų kombinacijos. Linijos ar poligono anotacijos braižomos kairiu pelės paspaudimu kiekvienos viršūnės vietoje, braižymą baigiant dešiniu pelės paspaudimu. Galima įjungti pritraukimą braižant, tikslioms viršūnių pozicijoms galite naudoti *Sudėtingesnio skaitmenizavimo įrankinę*, be to galite perjungti *braižymo įrankius* pavyzdžiui į tęstinį režimą visiškai laisvoms formoms.
	Kurti linijos anotaciją	
	Kurti žymeklio anotaciją	
Tekstas	Kurti teksto anotaciją taške	Spauskite žemėlapio drobėje kairiu mygtuku.
	Kurti tekstinę anotaciją palei liniją	Spauskite kairiu mygtuku, kad pridėtumėte viršūnes teksto bazinės linijos geometrijai, tada spauskite dešinį mygtuką, kad baigtumėte. Kaip aukščiau, galima naudoti geoobjektų skaitmeninimo galimybes.
	Kurti teksto anotaciją stačiakampyje	Brėžkite teksto stačiakampį, kairiu mygtuku padėkite pirmą kampą, tada vėl spauskite kairį mygtuką, kad pridėtumėte priešingą kampą.
Piešinys	Kurti paveikslo anotaciją	Brėžkite piešinio stačiakampį, kairiu mygtuku padėkite pirmą kampą, tada vėl spauskite kairį mygtuką, kad pridėtumėte priešingą kampą. Tada failų naršyklėje parinkite piešinį ar SVG failą.

Priešingai nei bendri sluoksniai, anotacijų sluoksnio nereikia aktyvuoti, kad galėtumėte pažymėti jo geoobjektus. Tiesiog čiupkite įrankį Keisti anotacijas ir galėsite sąveikauti su bet kuria anotacija:

- **Pažymėjimas**: spauskite kairį mygtuką ant anotacijos. Pagal nutylėjimą anotacijos braižomos sukūrimo tvarka, tai reiškia, kad paskutinės sukurtos anotacijos bus braižomos virš senesnių. Jums gali reikėti pažaisti su geoobjektų Z-index savybe, jei norite nurodyti, kas turi būti viršuje.
- **Perkelimas**: Spauskite kairę mygtuką ant pažymėtos anotacijos, kad pradėtumėte jos perkėlimą. Dešinys paspaudimas arba mygtukas `Esc` nutraukia perkėlimą, o antras kairys paspaudimas patvirtina perkėlimą. Perkėlimą taipogi galima valdyti ir rodyklių klavišais:
  - `Shift+key` dideliam perkėlimui
  - `Alt+key` for `1 tšk` perkėlimui
- **Geometrijos keitimas**: užveskite pelę virš anotacijos, ant po apačia esančios geometrijos taškų rodomi violetiniai kvadračiukai. Spauskite kairį mygtuką ant kvadrato, perkelkite, spauskite dar kartą. Linijų ar poligonų anotacijoms du kartus paspaudus ant segmento bus pridėta nauja viršūnė.
- **Ištrynimas**: Paspaudus `Del` ar `Backspace` mygtuką, kai anotacija pažymėta, ji bus ištrinta
- *Keisti geoobjekto simbologiją*

Geoobjekto simbologija

Pažymėtos anotacijos *Simbologijos* savybės bus rodomos skydelyje *Sluoksnio stilius*.

- Jūs galite keisti poligono, polilinijos ir žymeklio anotacijų išvaizdą naudodami visas *simbolių savybių* galimybes
- Teksto pagrindo anotacijoms, ploto su tekstu redagavimo įrankiai leidžia jums kurti rodomą tekstą. Jis gali būti kelių eilučių, naudoti HTML formatavimą ir naudoti QGIS išraiškų funkcijas. Be to, jūs galite keisti išvaizdą naudodami visas *teksto formatavimo savybių* galimybes.

  Priklausomai nuo anotacijos tipo galimos papildomos parinktys.
  - Teksto anotacijai taške taipogi galima:
    - nustatyti teksto *Lygiavimą* (kairėje, centre arba dešinėje nuo pririšimo taško)
    - konfigūruoti ar tekstas turi *Ignoruoti žemėlapio pasukimą* ar *Suktis su žemėlapiu*. Abiem atvejais galima nurodyti ir savo *Kampą* geoobjektų orientacijai.
  - Tekstinėms anotacijoms pagal liniją, konfigūruokite *Poslinkį nuo linijos* jūsų pasirinktais vienetais
  - Anotacijai stačiakampyje taipogi galima:
    - nustatyti teksto *Horizontalų lygiavimą* (kairėje, centre, dešinėje ar išlygiuotai) ir *Vertikalų lygiavimą* (viršuje, vertikaliame centre ar apačioje) stačiakampyje
    - konfigūruoti stačiakampio *Paraščių* atstumus jūsų pasirinkimo vienetais, *Kontūro* ir *Fono* spalvas.
- Piešinių anotacijos leidžia jums rodyti *Rastro piešinį* ar *SVG vaizdą*, naudojant *Nutolusio arba įtraukto failo parinkiklis*. Piešinio vietą ir dydį galima padaryti:
  - *Nuo mastelio priklausančio dydžio*: piešinio dydis kinta kartu su žemėlapiu, padidėja priartinant ir mažėja atitolinant. Dydis nurodomas naudojant vieną atskaitos dydį žemėlapio vienetais (pvz. metrais ar laipsniais, priklausomai nuo sluoksnio CRS). Naudojant varnelę *Užrakinti aspekto santykį*, jūs galite išlaikyti anotacijos dydį santykinai įtraukto piešinio dydžio arba ištempti piešinį, kad jis užpildytų anotacijos apimtį.
  - *Fiksuotas dydis*: piešinio anotacija išlaiko pastovų dydį ekrano ar išvesties vienetais (t.y. pikseliais ar milimetrais), nepriklausomai nuo žemėlapio mastelio. Anotacijos *Plotį* ir *Aukštį* galima pildyti atskirai arba keisti santykinai nuo įtraukto paveikslo dydžio.
  - *Santykinai žemėlapiui*: piešinio anotacija visada bus braižoma toje pačioje vietoje santykinai nuo žemėlapio drobės ribų, nepriklausomai nuo žemėlapio mastelio. Anotacijos *Plotį* ir *Aukštį* galima pildyti atskirai arba keisti santykinai nuo įtraukto piešinio dydžio.

  Dar galite įjungti stačiakampio *Kontūro* ir *Fono* rodymą bei konfigūruoti juos naudojant *pildymo simbolius*.

- Tekstui taške, tekstui stačiakampyje ir piešinių anotacijoms galite *Rodyti išnešimą*, kai paveikslas ar teksto pririšimo taškas yra patrauktas nuo pradinės vietos. Norėdami sukurti anotacijos išnešimą:
  1. Parinkite anotacijos elementą
  2. Perkelkite pririšimo tašką, pvz. spauskite centrinį ir žalią `X` tašką ir spauskite naujoje vietoje
  3. Anotacijų savybėse spauskite mygtuką … ir sukonfigūruokite *išnešimo savybes*.

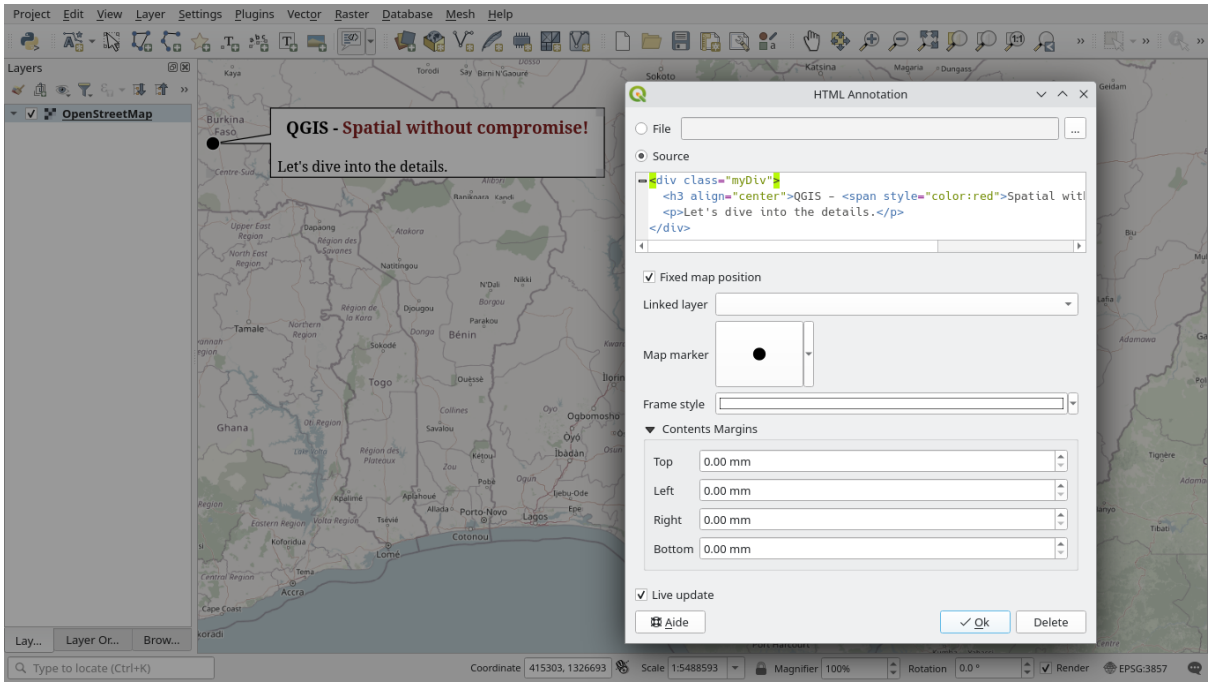
> **Patarimas**
>
> Pritaikius baliono išnešimą teksto ar piešinio anotacijai jūs gausite jūsų senąjį baliono anotacijos vaizdą.

- Konfigūruokite *Atskaitos mastelį*: nurodo žemėlapio mastelį, ties kuriuo simbolių ar teksto dydžiai naudoja popieriumi-paremtus vienetus (tokius kaip milimetrus ar taškus). Dydžiai bus atitinkamai pakeisti, kai žemėlapis bus žiūrimas skirtinguose masteliuose. Pavyzdžiui, 2mm pločio linijos geoobjektas 1:2000 *Atskaitos mastelyje* bus braižomas 4mm, kai žemėlapis žiūrimas mastelyje 1:1000.
- Nustatykite *Z-index*: geoobjektai su didesniu indeksu dedami virš geoobjektų su mažesniu indeksu. Patogus nustatymas tiek geoobjekto rodymui, tiek ir pažymėjimui.
- Keiskite kai kuriuos *Sluoksnio braižymo* nustatymus

Balionų anotacijos

Balionų anotacijas galite pridėti per meniu *Keisti* ► *Pridėti anotaciją* ► arba naudojant *Anotacijų įrankinė*:

- HTML Annotation to display a HTML-formatted string or a whole `HTML` file as an annotation.
- Form Annotation: useful to display attributes of a vector layer in a customized `.ui` file. This is similar to the *custom attribute forms*, but displayed in an annotation item.



7.13 Fig. : Example of a HTML balloon annotation

Kad pridėtumėte balionų anotaciją, parinkite atitinkamą įrankį ir spauskite žemėlapio drobėje. Pridedamas tuščias balionas. Du kartus jį paspauskite - atsivers dialogas su įvairiomis parinktimis. Šis dialogas beveik toks pat visiems

anotacijų tipams:

- *File*: Fill it with the path to a `.html` or `.ui` file, depending on the type of annotation, to serve as contents of the annotation.
- *Source*: Only available for HTML annotations, it allows to serve the annotation with a plain or HTML-formatted text instead of a file.
- *Fixed map position*: when unchecked, the balloon placement is based on a screen position (instead of the map), meaning that it's always shown regardless of the map canvas extent.
- *Susijęs sluoksnis*: susieja anotaciją su žemėlapio sluoksniu, todėl anotacija matoma tik tada, kai rodomas susietas sluoksnis.
- *Simbolis*: naudojant *QGIS simbolius*, nustato baliono susiejimo pozicijoje rodomą simbolį (rodomą tik kai įjungta parinktis *Fiksuota vieta žemėlapyje*).
- *Rėmo stilius*: nustato baliono rėmo fono spalvą, permatomumą, brėžimo spalvą ar plotį naudojant QGIS simbolius.
- *Turinio paraštės*: nustato vidines anotacijos rėmo paraštes.
- *Gyvas atnaujinimas* leidžia iš karto matyti jūsų pakeitimus.

Kai aktyvus baliono anotavimo įrankis, jūs taipogi galite:

- Pažymėti anotaciją
- Keisti anotacijos dydį
- Perkelti anotaciją į žemėlapio poziciją (pertempiant žemėlapio žymeklį) arba perkelti tik balioną.
- Ištrinti anotaciją: parinkite ją ir arba spauskite mygtuką `Del` ar `Backspace`, arba du kartus spauskite anotaciją ir savybių dialoge naudokite mygtuką *Trinti*.
- Paspaudus dešinį pelės mygtuką, per kontekstinį meniu galite:
  - *Kopijuoti koordinates* iš anotacijos žemėlapio žymeklio į skirtingas CRS
  - *Keisti* anotacijos savybes. Tas pats, kas du kartus paspaudus anotaciją.
  - *Trinti* anotaciją

7.1.8 Matavimas

Bendra informacija

QGIS teikia keturis geometrijų matavimo būdus:

- Interaktyvius matavimo įrankius
- matavimą naudojant Laukų skaičiuotuvą
- išvestinius matavimus identifikavimo įrankyje
- vektorinės analizės įrankius: *Vektorius* ► *Geometrijos įrankiai* ► *Pridėti geometrijos stulpelius*

Matavimas veikia tiek projektuotose koordinačių sistemose (pvz. UTM), tiek ir neprojektuotuose duomenyse. Pirmi trys matavimo įrankiai elgiasi pagal globalius projekto nustatymus:

- Kitaip nei dauguma GIS, numatytoji matavimo metrika yra elipsoidinė, naudojant elipsoidą, apibrėžtą *Projektas* ► *Savybės…* ► *Bendra*. Tai galioja tiek geografinėms, tiek ir projektuotoms sistemoms, apibrėžtoms projekte.
- Jei norite skaičiuoti projektuotą/planimetrinį plotą ar atstumą naudojant dekarto matematiką, matavimo elipsoidą reikia nurodyti „Jokio/Planimetrinis“ (*Projektas* ► *Savybės…* ► *Bendra*). Kaip ten bebūtų, nustačius duomenims ir projektui geografinę (t.y. neprojektuotą) CRS, ploto ir atstumo matavimai bus elipsoidiniai.

Nei identifikavimo įrankis, nei laukų skaičiuotuvas netransformuos jūsų duomenų į projekto CRS prieš matavimą. Jei norite tai pasiekti, jums reikės vektorinės analizės įrankio: *Vektorius* ► *Geometrijos įrankiai* ► *Pridėti geometrijos atributus…*. Čia matavimai bus planimetriniai, nebent pasirinksite elipsoidinį matavimą.

Interaktyvus ilgio, ploto, krypties ir kampo matavimas

Norėdami pradėti matavimą spauskite Atributų įrankinės piktogramą . Rodyklė žemyn greta piktogramos leidžia persijungti tarp ilgio, ploto, krypties ar kampo matavimo. Pagal nutylėjimą dialoge naudojamas vienetas, nurodytas meniu *Projektas* ► *Savybės…* ► *Bendra*.

Matuoti liniją ir the *Matuoti plotą* matavimai gali būti atliekami *Dekarto* arba *Elipsoidiniais* matavimais.

Pastaba

Matavimo įrankio konfigūravimas

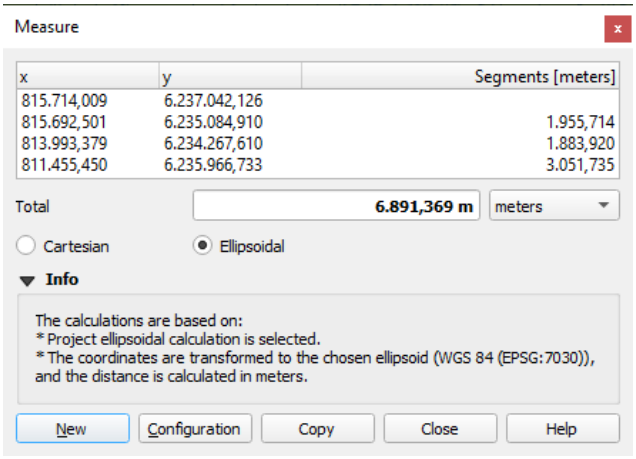
Matuojant ilgį arba plotą, paspaudus dialogo apačioje mygtuką *Nustatymai*, atidaromas meniu *Nustatymai* ► *Parinktys* ► *Žemėlapio įrankiai*, kur jūs galite parinkti gumelės spalvą, matavimų tikslumą bei vienetų elgseną. Taipogi galite parinkti norimus matavimo ar kampo vienetus, bet turėkite omenyje, kad šios reikšmės yra permušamos dabartinio projekto parinkime *Projektas* ► *Savybės…* ► *Bendra*, taip pat ir matavimo valdiklio valdiklio parinkimuose.

Visi matavimo moduliai naudoja skaitmeninimo modulio pritraukimo nustatymus (žr. skiltį *Setting the snapping tolerance and search radius*). Taigi jei norite tiksliai matuoti palei linijos geoobjektą arba aplink poligono geoobektą, visų pirma nustatykite to sluoksnio pritraukimo toleranciją. Tada, naudojant matavimo įrankius, kiekvienas pelės paspaudimas (tolerancijos nustatymo ribose) bus pritrauktas prie sluoksnio.

Matuoti liniją matuoja atstumą tarp nurodytų taškų. Įrankis leidžia jums spausti taškus žemėlapyje. Matavimo lange rodomas kiekvieno segmento ir bendras ilgis. Matavimo lange jūs matote visų taškų koordinates ir atstumus. Turėkite omenyje, kad pirma eilutė turės tik koordinates, nes ji reprezentuoja pradžios tašką. Dabar galima kopijuoti iš karto visus jūsų matavimus į iškarpinę naudojant mygtuką *Kopijuoti*. Paspaudę mygtuką *Konfigūracija* pasieksite *Matavimo įrankių kopijavimo nustatymai*, kur galėsite nustatyti *kopijavimo parinktis*. Norėdami baigti matavimą spauskite dešinį pelės mygtuką.

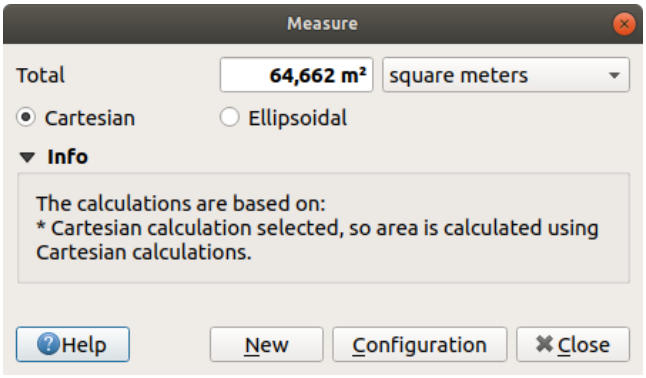
Pastebėtina, kad galite naudoti iškrentantį sąrašą, esantį greta bendros sumos, jei norite interaktyviai pakeisti *matavimo vienetus*, kol dirbate su matavimo įrankiu. Šie vienetai grąžinami valdiklyje kol sukuriamas naujas projektas arba atveriamas kitas projektas.

Dialogo skiltis *Informacija* paaiškina kaip vykdomi skaičiavimai pagal turimus CRS nustatymus.



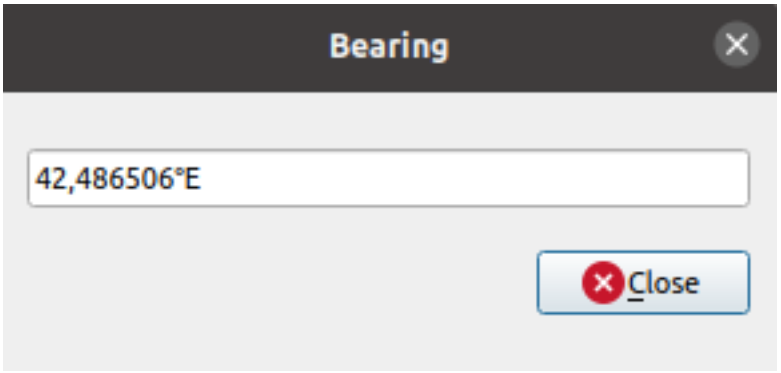
7.14 Fig. : Atstumo matavimas

Matuoti plotą: Galima matuoti ir plotus. Matavimo lange rodomas suminis plotas. Norėdami baigti braižyti spauskite dešinį mygtuką. Taipogi matoma ir informacijos skiltis bei yra galimybė perjungti skirtingus *matavimo vienetus*.



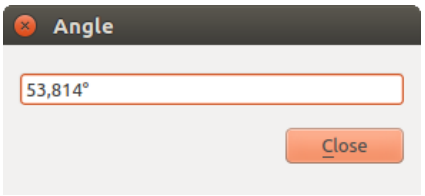
7.15 Fig. : Ploto matavimas

Matuoti kryptį: Jūs galite matuoti kryptis. Žymeklio forma pasikeičia į kryžių. Spauskite, kad nubrėžtumėte pirmą krypties tašką, tada perkelkite žymeklį, kad nubrėžtumėte antrą tašką. Matavimas rodomas iššokančiame dialoge.



7.16 Fig. : Krypties matavimas

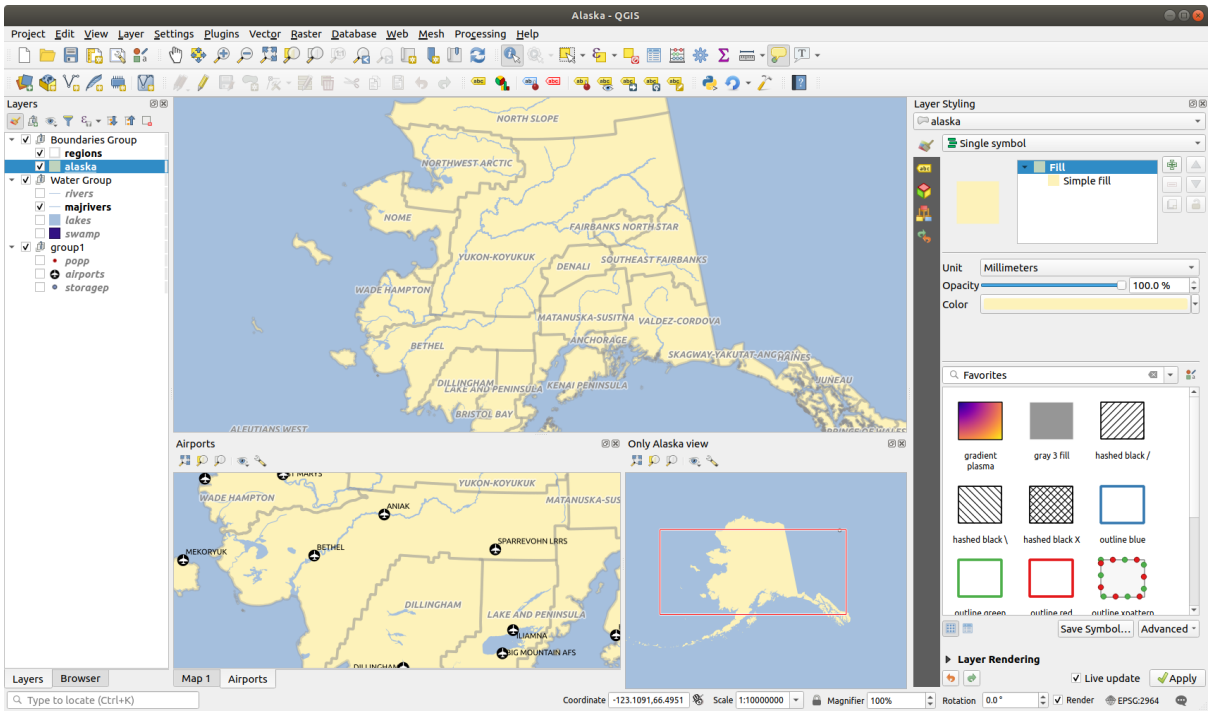
Matuoti kampą: Jūs taip pat galite matuoti kampus. Žymeklis tampa kryžiuku. Spauskite, kad nubrėžtumėte pirmą matuojamo kampo segmentą, tada tempkite žymeklį, kad nubraižytumėte norimą kampą. Matavimai rodomi iššokančiame dialoge.



7.17 Fig. : Kampo matavimas

7.1.9 Papildomų žemėlapio vaizdų kūrimas

Taipogi galima pridėti papildomus žemėlapio vaizdus, kurių turinys gali nukrypti nuo *Sluoksnių* skydelio esamos būsenos. Kad pridėtumėte naują žemėlapio vaizdą, eikite į *Rodymas* ► *Naujas žemėlapio vaizdas*. Į QGIS bus pridėtas naujas plaukiojantis valdiklis, atkartojantis pagrindinio žemėlapio vaizdą. Jūs galite pridėti tiek vaizdų, kiek tik norite. Juos galima palikti plaukioti, pridėti vieną greta kito ar vieną virš kito.



7.18 Fig. : Keli žemėlapio vaizdai su skirtingais nustatymais

Papildomos žemėlapio drobės viršuje yra įrankinė su tokiomis galimybėmis:

- Pilnai padidinti, Didinti iki pažymėjimo ir Didinti iki sluoksnio, navigacijai rodinyje
- Nustatyti rodymo temą vaizdo *žemėlapio temos* parinkimui. Nurodžius `(joks)`, vaizdas seks *Sluoksnių* skydelio pokyčius.
- Rodymo nustatymai žemėlapio vaizdo konfigūravimui:
  - *Sinchronizuoti rodinio centrą su pagrindiniu žemėlapiu*: sinchronizuoja žemėlapio vaizdų centrus nekeičiant mastelio. Tai leidžia turėti peržiūros stilių ar padidintą žemėlapį, kuris seka pagrindinės drobės centrą.
  - *Sinchronizuoti rodinį su pažymėjimu*: tas pats, kas priartinti pažymėjimą
  - *Mastelis*
  - *Pasukimas*
  - *Didinimas*
  - *Sinchronizuoti mastelį* su pagrindinio žemėlapio masteliu. *Mastelio daugiklis* gali būti pritaikytas, leidžiant jums pavyzdžiui turėti vaizdą, 2x stambesniu masteliu už pagrindinės drobės.
  - *Rodyti anotacijas*
  - *Rodyti kursoriaus poziciją*

- *Rodyti pagrindinės drobės apimtį*
- *Rodyti etiketes*: leidžia paslėpti etiketes nepriklausomai nuo to, ar jos įjungtos sluoksnio nustatymuose
- *Keisti žemėlapio CRS…*
- *Pervadinti rodinį…*

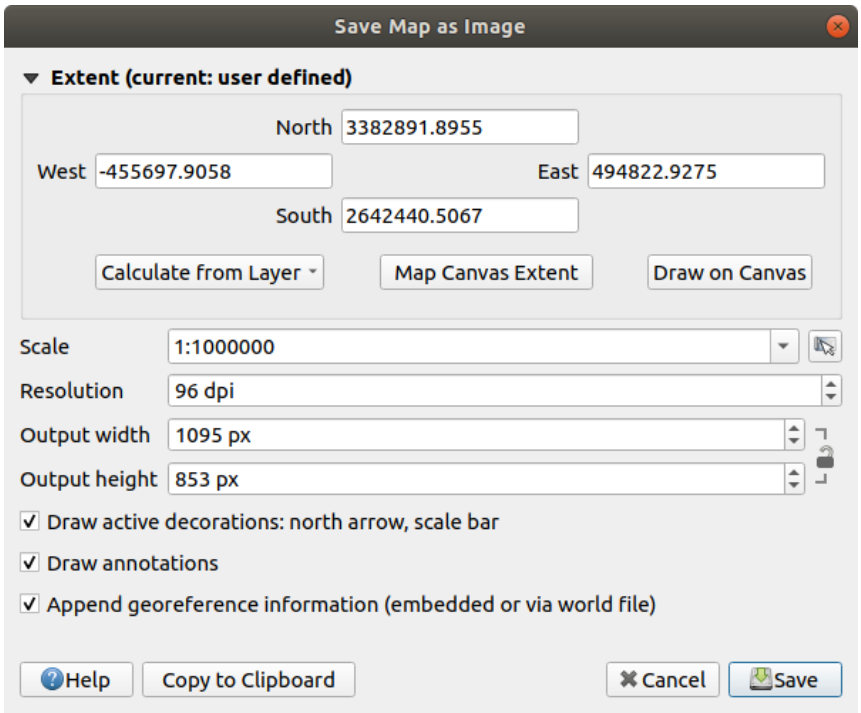
7.1.10 Žemėlapio vaizdo eksportavimas

Jūsų sukurti žemėlapiai gali būti išdėlioti ir eksportuoti į įvairius formatus naudojant išmaniąsias *spausdinimo išdėstymų ar ataskaitų* galimybes. Taipogi galima tiesiogiai eksportuoti šiuo metu nubraižytą vaizdą. Šis greitas žemėlapio vaizdo „fotografavimas“ turi keletą patogių savybių.

Eksportuoti dabartinio vaizdo žemėlapio drobę:

1. Eikite į *Proktas* ► *Importuoti/Eksportuoti*
2. Priklausomai nuo jūsų išvesties formato, parinkite arba
   - *Eksportuoti žemėlapį į piešinį…*
   - ar *Eksportuoti žemėlapį į PDF…*

Šie du įrankiai atsidariusiame dialoge suteikia jums pagrindines parinktis:



7.19 Fig. : Žemėlapio įrašymo į piešinį dialogas

1. Parinkite eksportuojamą *Apimtį*: tai gali būti dabar matomas vaizdas (numatytoji parinktis), sluoksnio apimtis, arba sava žemėlapyje nubraižyta apimtis. Parodomos parinkto ploto koordinatės ir jas galima keisti.
2. Įveskite žemėlapio *Mastelį* arba parinkite jį iš *iš anksto apibrėžtų mastelių*: pakeitus mastelį bus perskaičiuota eksporto apimtis (nuo centro).
3. Nustatykite išvesties *Rezoliuciją*

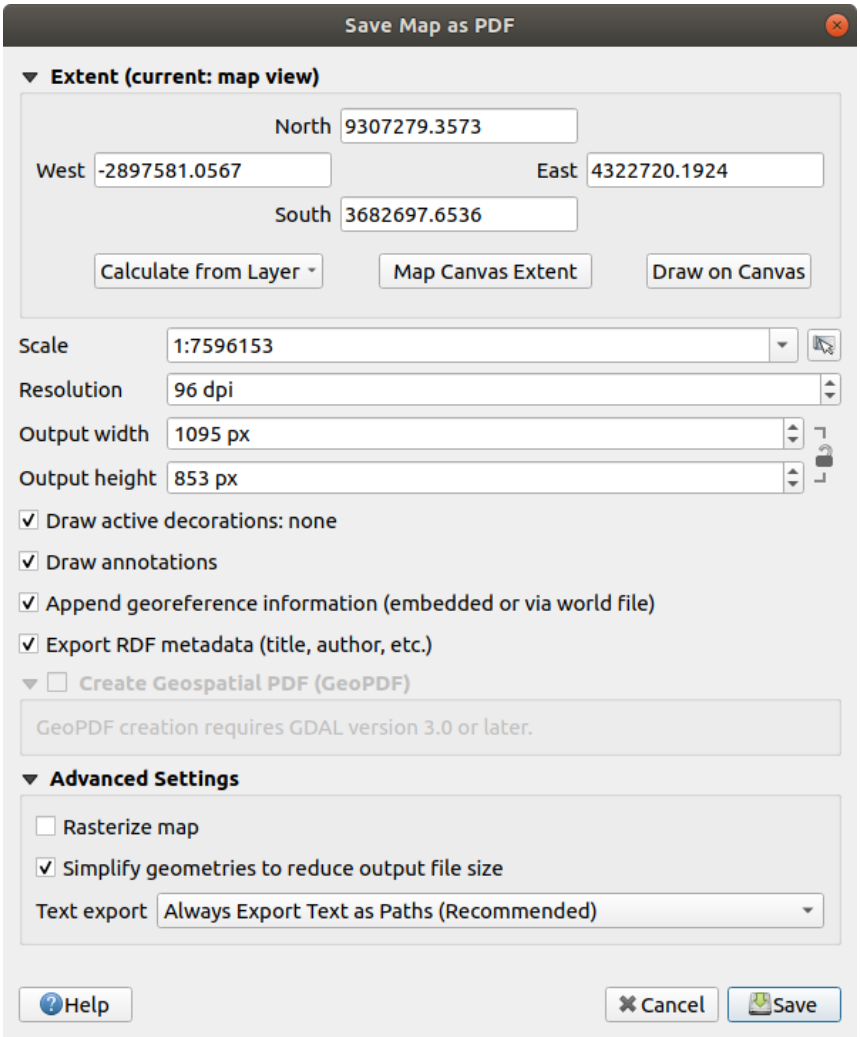
4. Valdykite *Išvesties plotį* ir *Išvesties aukštį* piešinio taškais: pagal nutylėjimą nurodoma pagal dabartinę rezoliuciją ir apimtį, bet gali būti pakeista ir tada pakeis žemėlapio apimtį (nuo centro). Dydžio santykis gali būti užrakintas, kas gali būti ypač patogu braižant apimtį drobėje.

5. ☑ *Paišyti aktyvias dekoracijas*: *dekoracijos* (mastelio juosta, pavadinimas, tinklelis, šiaurės rodyklė…) eksportuojamos kartu su žemėlapiu

6. ☑ *Paišyti anotacijas*, jei norite eksportuoti visas *anotacijas*

7. ☑ *Pridėti geopririšimo informaciją (įtraukta arba per pasaulio failą)*: priklausomai nuo išvesties formato, tame pačiame aplanke ir tokiu pačiu pavadinimu, kaip ir piešinys, bus įrašytas pasaulio failas (su priesaga `PNGW` `PNG` piešiniams, `JPGW` `JPG`, …). `PDF` formate ši informacija įdedama į PDF failo vidų.

8. Eksportuojant į PDF pateikiama daugiau parinkčių *Įrašyti žemėlapį į PDF…* dialoge:



7.20 Fig. : Žemėlapio įrašymo į PDF dialogas

- ☑ *Eksportuoti RDF metaduomenis* - pridedami dokumento duomenys, tokie kaip pavadinimas, autorius, data, aprašymas…
- ☐ *Kurti geoerdvinį PDF*: Sukuria geografiškai priristą PDF failą. Jūs galite:
  - Parinkti Geoerdvinį PDF *Formatą*
  - ☑ *Įtraukti vektorinių geoobjektų informaciją* Geoerdvinį PDF faile: įtrauks visus geometrijų atributus iš geoobjektų, matomų išvesties Geoerdvinį PDF faile.

> **Pastaba**
>
> Geoerdvinį PDF failą taipogi galima naudoti kaip duomenų šaltinį. Daugiau informacijos apie Geoerdvinių PDF palaikymą QGIS'e, žiūrėkite https://north-road.com/2019/09/03/qgis-3-10-loves-geospatialpdf/.

- *Rastrizuoti žemėlapį*
- *Paprastinti geometrijas failo dydžio sumažinimui*: Geometrijos bus supaprastintos pašalinant viršūnes, kurios nekeičia vaizdo eksporto rezoliucijoje (pvz. jei eksporto rezoliucija yra `300 dpi`, viršūnės, kurios yra arčiau viena kitos nei `1/600 colio`, bus pašalintos). Tai gali sumažinti eksportuojamo failo dydį ir sudėtingumą (labai didelių failų kai kurios programos gali nesugebėti atidaryti).
- Nustatymas *Teksto eksportas*: valdo ar teksto užrašai visada ar pageidaujamai eksportuojami kaip *teksto ar kontūro objektai*.

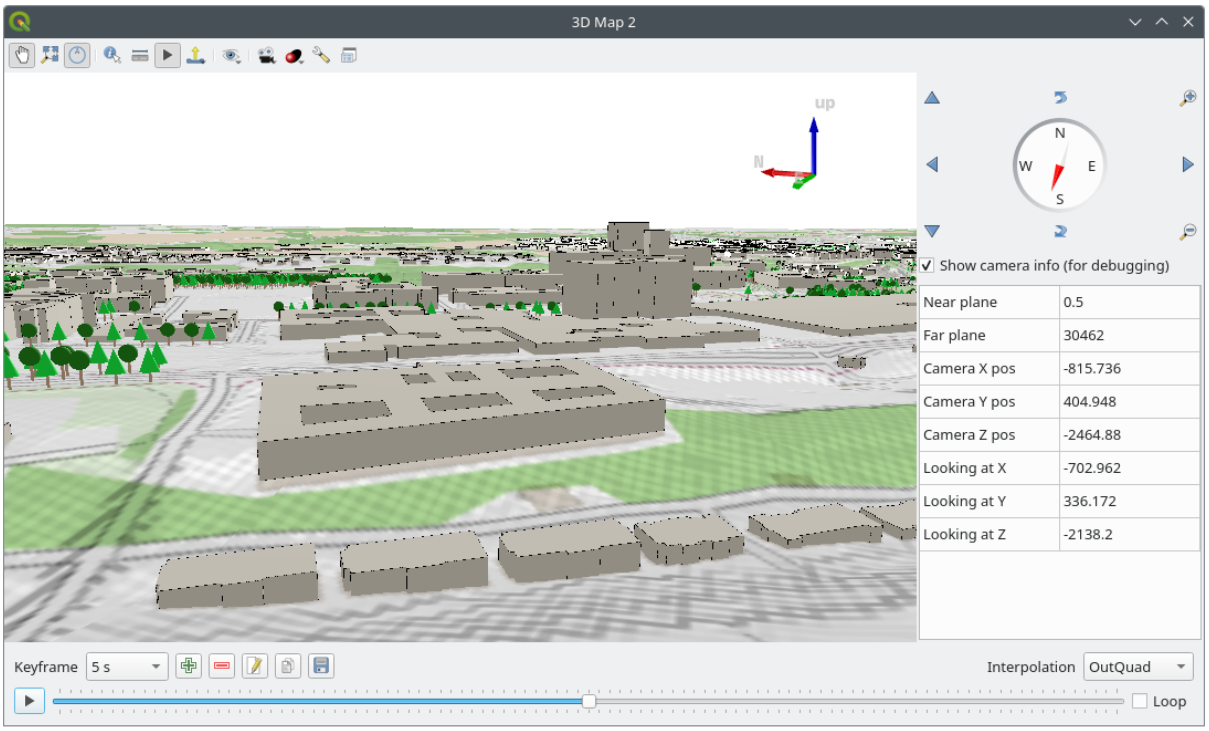
9. Spauskite *Įrašyti* ir parinkite failo vietą, pavadinimą ir formatą.

   Eksportuojant į piešinį galime *Kopijuoti į iškarpinę* aukščiau parinktų parametrų rezultatą, o tada kitoje aplikacijoje, tarkim LibreOffice ar GIMP, tiesiog jį įkelti…

7.2 3D žemėlapio vaizdas

3D vizualizavimas palaikomas 3D žemėlapio vaizde. Jūs galite sukurti, tvarkyti ir atverti 3D žemėlapio vaizdus per meniu *Rodymas* ► *3D žemėlapio vaizdai* ►:

1. Spausdami *Naujas 3D žemėlapio vaizdas* jūs galite sukurti naują 3D žemėlapio vaizdą. Pasirodys plaukiojantis ir prijungiamas QGIS skydelis (žr. *3D žemėlapio vaizdo dialogas*). Jis turi tokią pačią apimtį ir vaizdą kaip ir pagrindinė 2D žemėlapio drobė ir teikia rinkinį įrankių, leidžiančių keisti vaizdą į 3D.
2. Spausdami *Valdyti 3D žemėlapio vaizdus* jūs galite patekti į 3D žemėlapio vaizdų tvarkyklę. Čia turite galimybę atverti, dubliuoti, pašalinti ar pervadinti 3D žemėlapių vaizdus.
3. Jei sukuriate vieną ar daugiau 3D žemėlapio vaizdų, jūs matote juos išvardintus meniu *3D žemėlapio vaizdai*. Paspaudę ant jų galite juos įjungti ir išjungti. Jie bus įrašyti kartu su projektu, net jei jie išjungti.



7.21 Fig. : 3D žemėlapio vaizdo dialogas

Šie įrankiai teikiami 3D žemėlapio vaizdo skydelio viršuje:

- Kameros valdymas: keičia vaizdą, paliekant tą patį kampą ir kameros kryptį
- Pilnai padidinti: keičia vaizdo dydį taip, kad tilptų visa sluoksnio apimtis, arba *atskaitos apimtis*, jei ji nurodyta
- Perjungti ekrano pranešimus: rodo/slepia navigacijos valdiklį (tai skirta palengvinti žemėlapio vaizdo valdymą)
- Identifikuoti: grąžina informaciją apie paspaustą tašką paviršiuje arba paspaustą 3D geoobjektą(us) – daugiau informacijos rasite skyriuje *Geoobjektų identifikavimas*
- Measurement Line: measures the horizontal, vertical, and total 3D distance between points
  - Click to add points
  - Right-click to finish the measurement
  - Right-clicking twice or pressing *New* will reset the table
- Animacijos: rodo/slepia *animacijos grotuvo* valdiklį
- Eksportuoti leidžia jums eksportuoti sceną į įvairius formatus:
  - Įrašyti kaip piešinį…: įrašo dabartinį vaizdą kaip piešinio formato failą
  - Eksportuoti 3D sceną: eksportuoja dabartinį vaizdą kaip 3D sceną (`.obj` failą), leidžia vėliau apdoroti programose, tokiose kaip Blender… Paviršiaus ir vektoriniai geoobjektai eksportuojami kaip 3D objektai. Eksporto nustatymai, permušantys sluoksnio *savybes* ar žemėlapio vaizdo *konfigūraciją*, yra:
    - *Scenos pavadinimas* ir paskirties *Aplankas*
    - *Paviršiaus rezoliucija*
    - *Paviršiaus tekstūros rezoliucija*

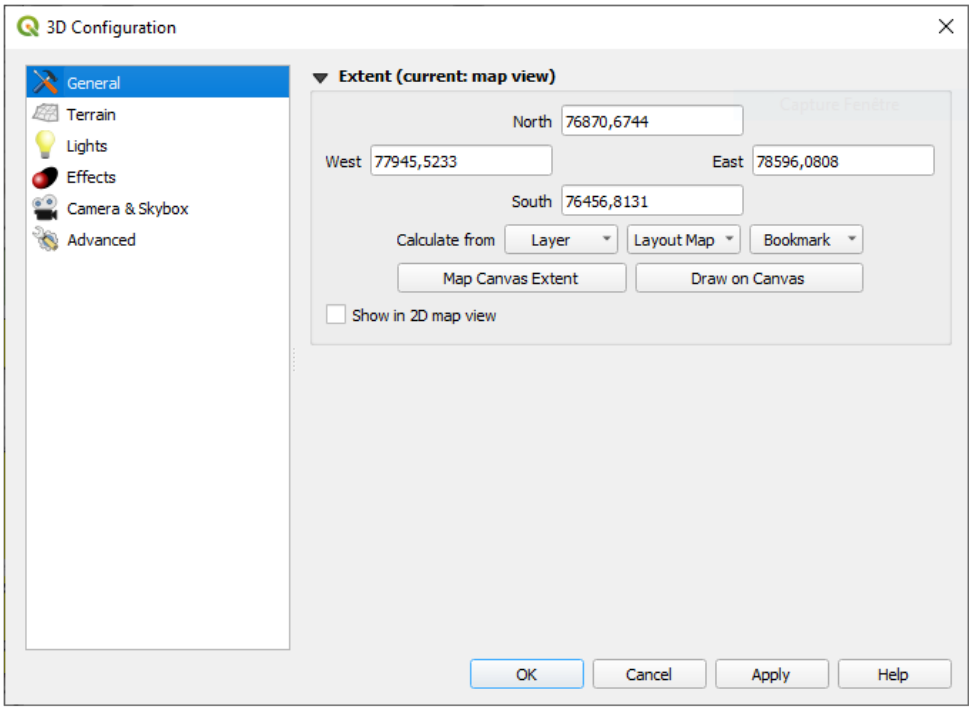
    - *Modelio mastelis*
    - *Glotninti kampus*
    - *Eksportuoti normales*
    - *Eksportuoti tekstūras*
- Nustatyti vaizdo temą: Leidžia jums parinkti ir nustatyti sluoksnius, kurie bus rodomi žemėlapio vaizde, pagal iš anksto nustatytas *žemėlapio temas*.
- Meniu Kamera padeda jums valdyti ryšį tarp 2D ir 3D vaizdo:
  - Sinchronizuoti vaizdus (*2D žemėlapio vaizdas seka 3D kamerą* ir/arba *3D kamera seka 2D žemėlapio vaizdą*)
  - *Rodyti kameros matomą plotą 2D žemėlapio vaizde*
  - *Nustatyti 3D sceną 2D žemėlapio vaizde*: leidžia apkirpti 3D sceną ir rodyti tik paviršių ir geoobjektus, kurie kertasi su 2D žemėlapio drobėje apibrėžta apimtimi. Daugiau parinkčių galima rasti kortelėje *Bendra konfigūracija*.
  - Cross Section Tool creates a cross section in 3D scene by defining an area on the terrain in the 2D map canvas. Click to set the start point, move the mouse to define the line direction and click again to set the end point. You can adjust the size of the section by moving the mouse after selecting the start and end points. Press `Esc` or right-click to reset the selection.
  - Use Disable Cross Section to remove the cross section.
- Efektai prideda į 3D braižymą vizualius efektus, tokius kaip *šešėliai*, *akių kupolo apšvietimą* ar *aplinkos okliuziją*.
- The Configure… button opens the dialog to configure the 3D map view *settings*.
- Prisegti 3D žemėlapio vaizdą: persijungti tarp prisegto valdiklio ir aukščiausio lygio lango

7.2.1 Scenos konfigūravimas

The 3D map view opens with some default settings you can customize. To do so, press the Configure… button at the top of the 3D canvas panel to open the *3D configuration* window.

3D konfigūracijos lange yra keletas parinkčių, kuriomis galite tinkinti 3D sceną:

Bendra



7.22 Fig. : 3D žemėlapio bendros konfigūracijos dialogas

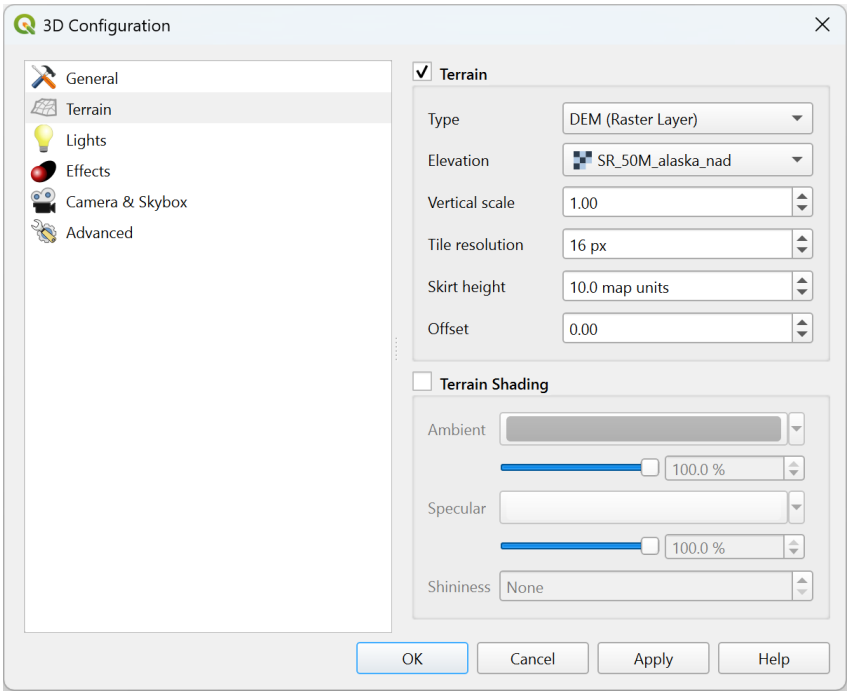
Kortelėje *Bendra*, jūs galite:

- Riboti 3D scenos turinį 2D žemėlapio apimtimi naudojant *erdvinės apimties parinkiklį*: paviršius ir geoobjektai apkarpomi ties nurodyta apimtimi ir į sceną įkeliamos tik tos dalys, kurios patenka į apimtį. Kad kamera nebūtų pastatyta žemiau už scenos turinį, atsižvelgiama į paviršiaus, tinklelio ar taškų debesies aukščio diapazoną, ypač naudojant paviršiau vertikalaus mastelio nustatymą siekiant padidinti aukščio skirtumus.

  Aibės apimtis taipogi naudojama kaip atskaitos apimtis, kai 3D žemėlapio vaizde spaudžiamas mygtukas Pilnai padidinti.

- Įjunkite *Rodyti 2D žemėlapio vaizde*, kad dabartinė 3D scenos apimtis būtų rodoma pagrindinėje žemėlapio drobėje kaip gumelė.

Paviršius



7.23 Fig. : 3D žemėlapio paviršiaus konfigūravimo dialogas

- *Paviršius*: Prieš neriant į detales, svarbu pastebėti, kad paviršius 3D vaizde reprezentuojamas paviršiaus kaladėlių hierarchija ir kai kamera priartėja prie paviršiaus, esamos kaladėlės, kuriose nebėra pakankamai detalių, pakeičiamos mažesnėmis kaladėlėmis su daugiau detalių. Kiekvienoje kaladėlėje yra tinklo geometrija, gauta iš aukščio rastro sluoksnio ir 2D žemėlapių sluoksnių tekstūros.
  - Aukščio paviršiaus *Tipai* gali būti:
    - *Plokščias paviršius*
    - įkeltas *DEM (Rastro sluoksnis)*
    - *Internetinė* paslauga, įkelianti aukščio kaladėles, sukurtas Mapzen įrankiais – daugiau informacijos https://registry.opendata.aws/terrain-tiles/
    - įkeltas *Tinklelio* duomenų rinkinys
    - *Kvantizuoto tinklelio* sluoksnis iš projekto
  - *Aukščio*: rastras ar tinklelio sluoksnis, kurie naudojami paviršiaus kūrimui. Rastro sluoksnyje turi būti aukščio juosta. Tinklelio sluoksnio atveju naudojamos viršūnių Z reikšmės.
  - *Vertikalus mastelis*: Mastelio faktorius vertikaliam masteliui. Didinant šį mastelį bus papildomai didinamas paviršiaus aukštis.
  - *Kaladėlių rezoliucija*: Kiek paviršiaus rastro mėginių naudoti kiekvienai kaladėlei. Reikšmė 16px reiškia, kad kiekvienos kaladėlės geometrijoje bus 16x16 aukščio mėginių. Didesni skaičiai reiškia detalesnį paviršių sudėtingesnio braižymo kaina.
  - *Sijono aukštis*: Kartais galima matyti mažus tarpelius tarp paviršiaus kaladėlių. Didinant šią reikšmę, aplink paviršiaus kaladėles bus pridėtos vertikalios sienos („sijonai“), slepiančios tarpelius.
  - *Poslinkis*: pastumia paviršių aukštyn ar žemyn, pvz. keičiant aukštį kitų scenos objektų santykyje su žeme.

    Taii gali būti naudinga, kai yra nesutapimas tarp paviršiaus aukščių ir sluoksnių aukšio jūsų scenoje (pvz. taškų debesys, kurie naudoja tik santykinius vertikalius aukščius). Tokiu atveju rankiniu būdu keičiant

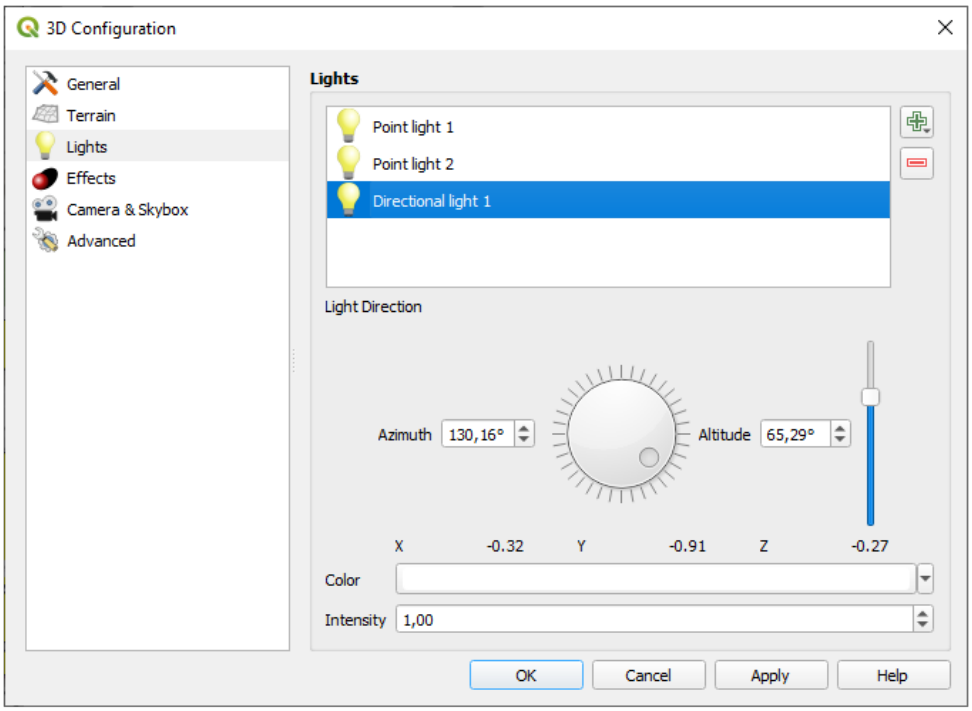
paviršiaus aukštį galima juos sutapatinti su jūsų scenos objektų aukščiu taip pagerinant navigacijos patirtį.

- Kai tinklelio sluoksnis naudojamas kaip paviršius, jūs galite keisti *Trikampių nustatymus* (tinklelio rodymą, glotnius trikampius, detalumo lygį) ir *Braižymo spalvų nustatymus* (kaip vieną spalvą ar *spalvų rampą*). Daugiau informacijos rasite skiltyje *Tinklelio sluoksnio 3D savybės*.
- *Paviršiaus pildymas*: Leidžia jums pasirinkti, kaip bus braižomas paviršius:
  - Pildymas išjungtas - paviršiaus spalvą nustato tik žemėlapio tekstūra
  - Pildymas įjungtas - paviršiaus spalva nustatoma naudojant Phong pildymo modelį, atsižvelgiant į žemėlapio tekstūrą, paviršiaus normalų vektorių, scenos apšvietimą, paviršiaus medžiagą, *Pasyvias* ir *Atsipindinčias* spalvas bei *Blizgumą*

Šviesos

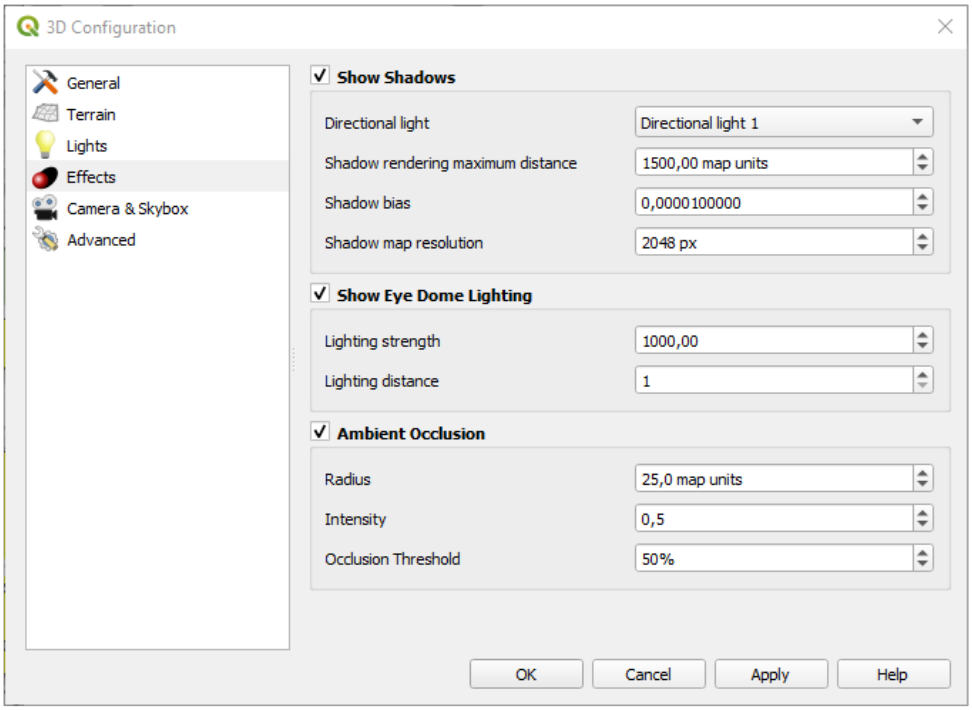
Kortelėje *Šviesos*, spauskite meniu, kad pridėtumėte

- iki aštuonių *Taškinių šviesų*: skleidžia šviesą visomis kryptimis, kaip šviesos sfera, užpildanti plotą. Objektai, kurie yra arčiau šviesos bus ryškesni, o toliau esantys objektai - tamsesni. Taškinė šviesa turi konkrečią poziciją (*X*, *Y* ir *Z*), *Spalvą*, *Intensyvumą* ir *Slopinimą*
- iki keturių *Kryptinių šviesų*: atkartoja šviesą, kurią gautumėte iš didžiulio šviesos šaltinio, esančio toli nuo jūsų objektų, visada centruotą ir niekada netemstančią (pvz. Saulę). Ji skleidžia paralelinius šviesos spindulius viena kryptimi ir šviesa pasiekia begalybę. Kryptinė šviesa gali būti pasukta nurodžius *Azimutą*, turi *Aukštį*, *Spavą* ir *Intensyvumą*.



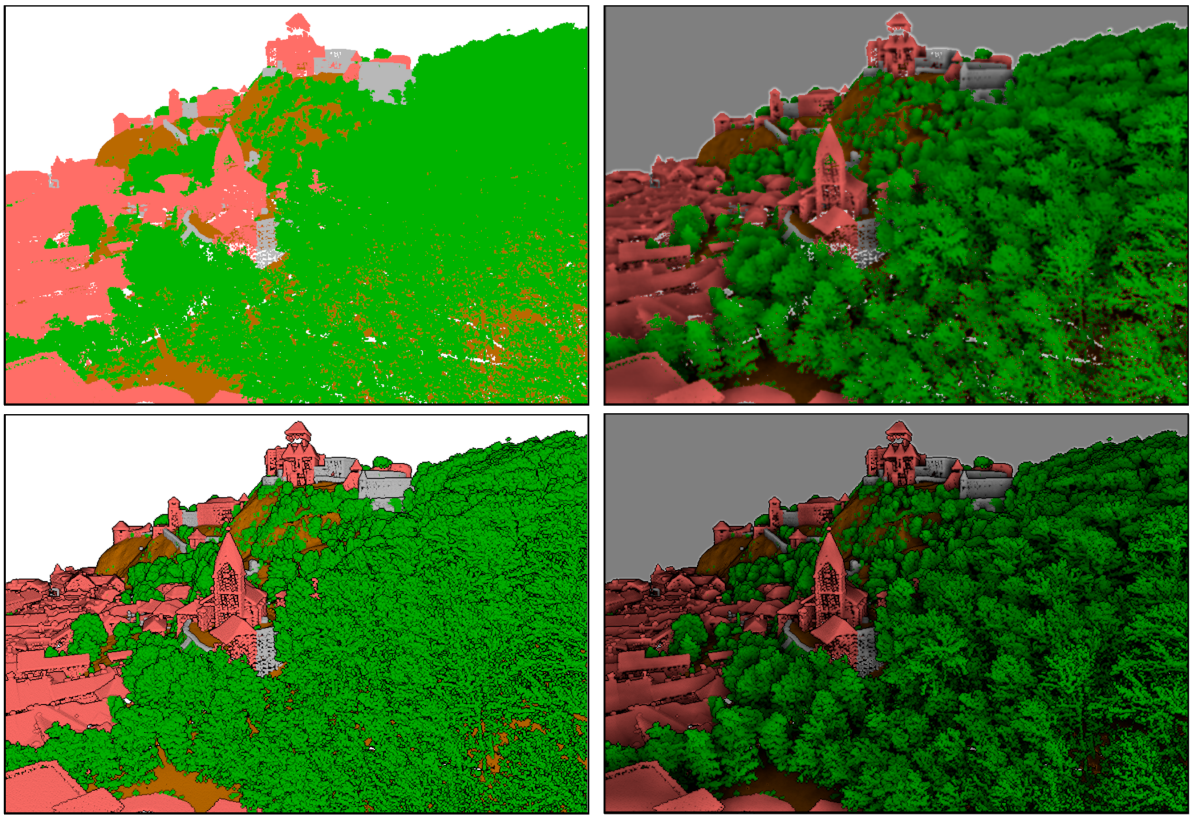
7.24 Fig. : 3D žemėlapio šviesų konfigūravimo dialogas

Efektai



7.25 Fig. : 3D žemėlapio efektų konfigūravimo dialogas

- Įjunkite parinktį *Rodyti šešėlius*, kad jūsų scenoje rodytumėte šešėlius, pagal nurodytus:
  - *Kryptinę šviesą*
  - *Šešėlio braižymo maksimalų atstumą*: kad būtų išvengta šešėlių braižymo per doli esantiems objektams, ypač kai kamera žiūri į horizontą
  - *Šešėlių šališkumą*: kad būtų išvengta savo šešėlių efektų, kurie kai kuriuos plotus gali padaryti tamsesniais už kitus, dėl žemėlapio dydžių skirtumų. Kuo mažesnis, tuo geriau
  - *Šešėlio žemėlapio rezoliucija*: kad šešėliai būtų aštresni. Tai gali reikšti prastesnę greitaveiką, jei rezoliucijos parametras per aukštas.
- *Rodyti akių kupolo apšvietimą* (EDL): po apdorojimo einantis efektas, kuris pagerina gylio suvokimą. Kiekvieno taško gylis (atstumas nuo kameros) lyginamas su kaimyninių taškų gyliu ir paryškinamas pagal gylio skirtumą, taip išskiriant kraštus. Įtakoja visą sceną ir gali būti kombinuojamas su *Ekrano erdvės aplinkos okliuzija*. Galima valdyti šiuos parametrus:
  - *Apšvietimo stiprumas*: didina kontrastą, taip leidžiant geresnį gylio suvokimą
  - *Apšvietimo atstumas*: reprezentuoja naudotų taškų atstumą nuo taško centro ir padaro kraštus storesniais.
- Pridėti ekrano erdvės *Aplinkos okliuziją* (SSAO): efektas po apdorojimo, kuris taipogi pagerina gylio suvokimą pridedant tamsesnį atspalvį plotams, kurie yra mažiau paveikti aplinkos šviesos. Įtakoją visą sceną ir gali būti kombinuojama su *Akių kupolo apšvietimu*. Galima valdyti šiuos parametrus:
  - *Spindulys*: kiek toli mes pasieksime skaičiuodami aplinkos okliuziją
  - *Intensyvumas*: kiek stiprus turi būti efektas (didesnės reikšmės daro daiktus tamsesniais)
  - *Okliuzijos slenksnis*: kiek kaimyninių taškų turi būti užkimšti, kad pradėtų matytis efektas (mažesnės už 50% reikšmės rezultatą padarys tamsesniu, bet tikėtina, kad pateiks didesni okliuzijos diapazoną)

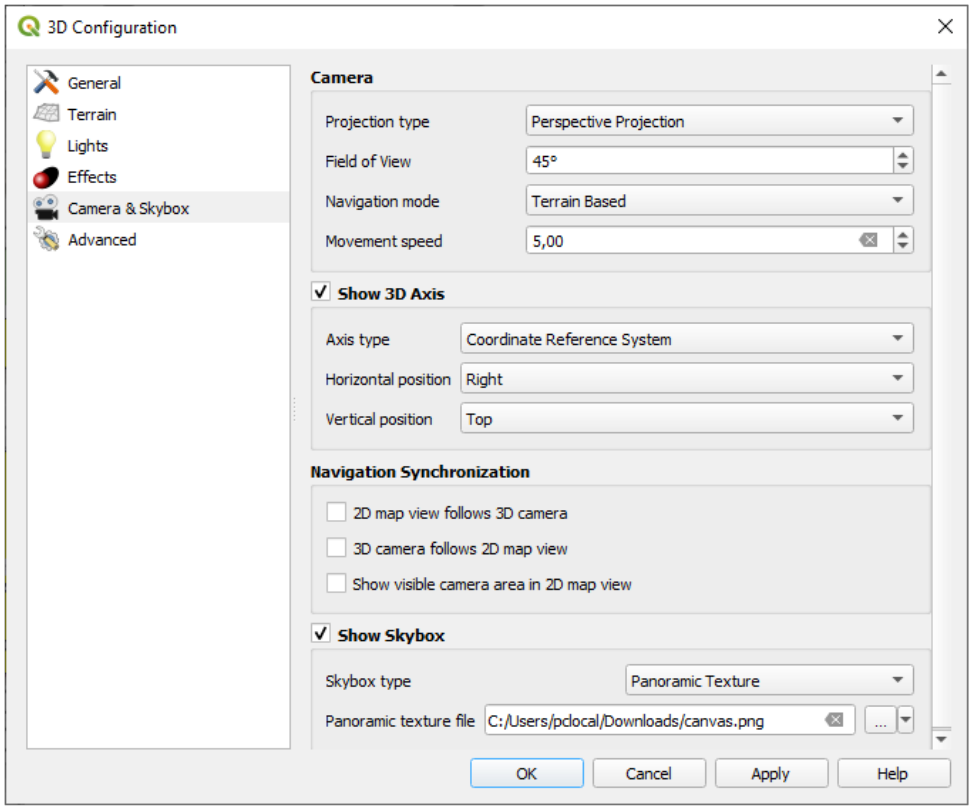


7.26 Fig. : Taškų debesų braižymas 3D žemėlapyje naudojant Akių kupolo apšvietimo (EDL) ir/ar ekrano erdvės aplinkos okliuziją (SSAO)

Nuo viršaus, iš kairės į dešinę: Be efektų – tik SSAO – tik EDL – SSAO ir EDL

Kamera ir dangus

Šioje kortelėje jūs galite valdyti įvairius parametrus, tokius kaip kamerą, 3D ašis, navigacijos sinchronizaciją ir dangų.

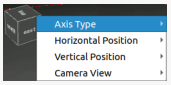


7.27 Fig. : 3D žemėlapio kameros konfigūravimo dialogas

- *Kameros* parametrų grupė permuša kai kuriuos *numatytuosius kameros nustatymus*, padarytus dialoge *Nustatymai* ► *Parinktys* ► *3D*.
- Įjunkite parinktį *Rodyti 3D ašis*, kad įjungtumėte 3D ašies įrankį. Ši parametrų grupė leidžia nustatyti ašies tipą ir jos poziciją.
  - Nurodžius tipą *Koordinačių atskaitos sistema*, bus rodoma statmena ašis.
  - Nurodžius tipą *Kubas*, bus rodomas 3D kubas. Kubo paviršius galima naudoti kameros vaizdo keitimui: pavyzdžiui spauskite ant *šiaurės* paviršiaus, jei norite, kad kamera matytų iš šiaurės.

Patarimas

Spauskite ant 3D ašies, kad greitai nustatytumėte poziciją ir tipą bei kameros vaizdą.

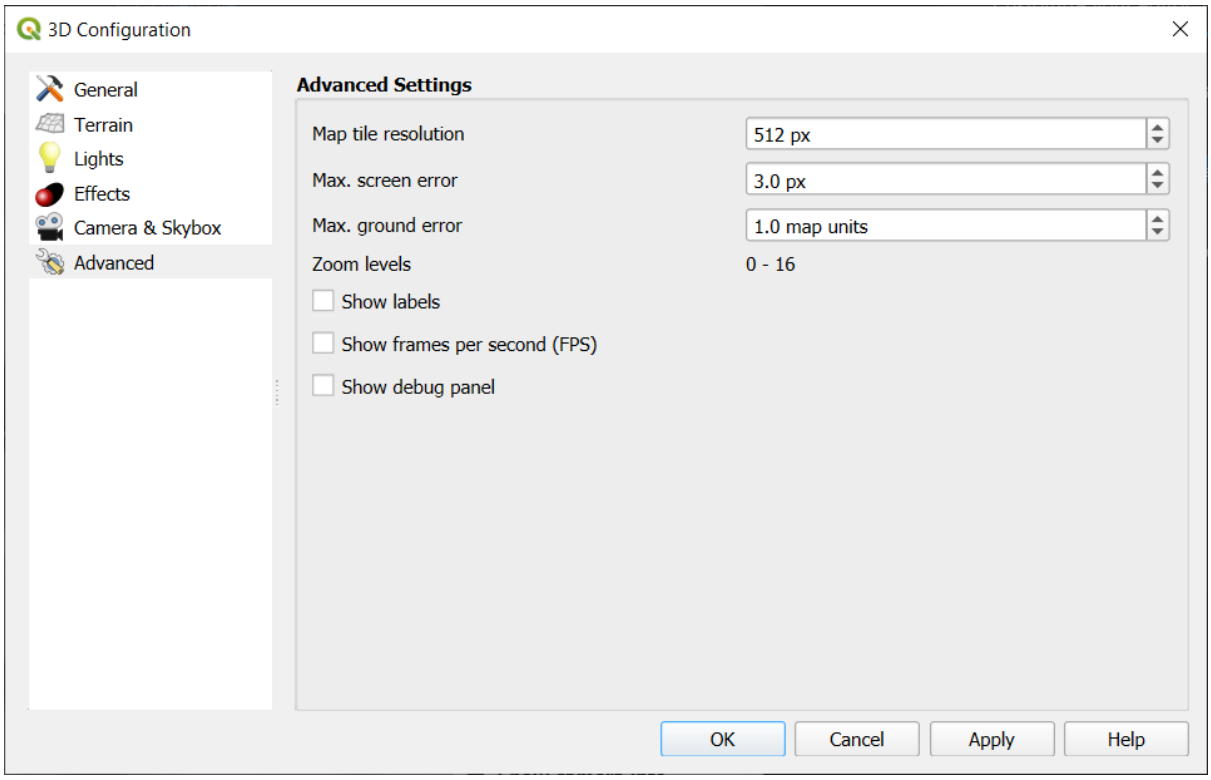


7.28 Fig. : 3D ašies kontekstinis meniu

- *Navigacijos sinchronizavimo* parametro grupėje yra parinktys 2D vaizdo sinchronizavimui su 3D kameros pozicija arba 3D kameros pozicijai su 2D vaizdu arba abiejų krypčių sinchronizacijai. Paskutinė parinktis rodo 3D kameros matomą apimtį 2D žemėlapio vaizde.
- Įjunkite *Rodyti dangų*, jei norite įjungti scenoje dangaus braižymą. Galimi dangaus tipai:
  - *Panoraminė tekstūra*, su vienu failu, teikiančiu 360° vaizdą
  - *Atskiri paviršiai*, su tekstūros failu kiekvienai iš šešių scenos dėžutės pusių

Dangaus tekstūros nuotraukos gali būti failai diske, nutolę URL arba į projektą įdėti failai (*daugiau informacijos*).

Sudėtingesni



7.29 Fig. : 3D žemėlapio sudėtingesnio konfigūravimo dialogas

- *Žemėlapio kaladėlių rezoliucija*: 2D žemėlapio piešinių, naudojamų kaip paviršiaus tekstūros, plotis ir aukštis. 256px reiškia, kad kiekviena kaladėlė bus braižoma į 256x256 pikselių piešinį. Didesni skaičiai sukuria detalesnes paviršiaus kaladėles sudėtingesnio braižymo kaina.
- *Maks. ekrano klaida*: Nurodo slenkstį keičiant paviršiaus kaladėles detalesnėmis (ir atvirkščiai) - pvz. kaip greitai 3D vaizdas naudos aukštesnės kokybės kaladėles. Mažesni skaičiai reiškia daugiau detalių scenoje padidėjusio braižymo sudėtingumo kaina.
- *Maks. žemės klaida*: Paviršiaus kaladėlių rezoliucija, kurioje kaladėlių dalinamas į detalesnes bus stabdomas (dalinant jas vis tiek nebebūtų gaunama papildomų detalių). Ši reikšmė riboja kaladėlių hierarchijos gylį: mažesnės reikšmės reiškia gilesnę hierarchiją didinant braižymo sudėtingumą.
- *Mastelio lygiai*: Rodo mastelio lygių skaičių (priklauso nuo žemėlapio kaladėlių rezoliucijos ir maksimalios žemės klaidos).
- ☐ *Rodyti etiketes*: Perjungia žemėlapio etikečių rodymą
- ☐ *Rodyti kadrus per sekundę (FPS)*
- ☐ *Show debug panel*: allows to show or hide a panel next to the 3D map view displaying options to better control map navigation and debug information:
  - ☐ *Rodyti žemėlapio kaladėlės info*: įtraukti paviršiaus kaladėlių rėmelį ir kaladėlės numerius (naudinga diagnozuojant paviršiaus problemas)
  - ☐ *Rodyti apimties stačiakampius*: Rodyti paviršiaus kaladėlių 3D apimties stačiakampius (naudinga diagnozuojant paviršiaus problemas)

- ☐ *Rodyti kameros vaizdo centrą*
- ☐ *Rodyti kameros sukimo centrą*
- ☐ *Rodyti šviesos šaltinius*: rodo sferą šviesos šaltiniuose, leidžiant lengviau padėti ir keisti šviesos šaltinius santykinai scenos turiniui
- ☐ *Stop scene updates*
- ☐ *Show debug overlay*: visual overlay which displays some useful debugging and profiling information. This allows in particular to quickly see the frame graph and the scene graph
- ☐ *Tinkinti šešėlių žemėlapį*: braižo sceną kaip raudonai-juodą piešinį iš šešėliams naudojamos šviesos perspektyvos (problemų diagnozei). Valdiklis nustatytas su proporcingu *Dydžiu* pagal 3D žemėlapio vaizdą ir prisegtas *Kampe*.
- ☐ *Tinkinti gylio žemėlapį*: braižo scenos gylio žemėlapį kaip piešinį, kuriame artimesni taškai yra tamsesni (problemų diagnozei). Valdiklis nustatytas su proporcingu *Dydžiu* pagal 3D žemėlapio vaizdą ir prisegtas *Kampe*.
- *Show camera info*:
  - ☐ *Far plane*: controls how far from the camera rendering stops
  - ☐ *Near plane*: controls how close to the camera rendering starts
  - ☐ *Camera X/Y/Z pos*: sets the camera's position in 3D space
  - ☐ *Looking at X/Y/Z*: sets the target point the camera is looking at

Pastaba

When your 3D map view is open in a standalone window, you can use `Ctrl+Shift+d` keyboard shortcut to access the debug panel.

7.2.2 Navigacijos parinktys

Žemėlapio vaizdo tyrinėjimui 3D:

- Palenkti paviršių (pasukant jį aplink horizontalią ašį, einančią per lango centrą)
  - Spauskite įrankius Sukti aukštyn ir Sykti žemyn
  - Laikykite `Shift` ir naudokite mygtukus aukštyn/žemyn
  - Tempkite pelę pirmyn/atgal nuspaudę vidurinį mygtuką
  - Laikykite `Shift` ir tempkite pelę pirmyn/atgal su kairiu pelės mygtuku
- Pasukti paviršių (aplink vertikalią ašį, kuri eina per lango centrą)
  - Pasukite navigacijos valdiklio kompasą žiūrėjimo kryptimi
  - Laikykite `Shift` ir naudokite mygtukus kairėn/dešinėn
  - Tempkite pelę dešinėn/kairėn nuspaudę vidurinį mygtuką
  - Laikykite `Shift` ir tempkite pelę dešinėn/kairėn nuspaudę kairį mygtuką
- Keiskite kameros poziciją (ir vaizdo centrą) tempdami jį horizontalioje plokštumoje
  - Tempkite pelę laikydami kairį mygtuką įjungę mygtuką Kameros valdymas
  - Spauskite krypties rodykles navigacijos valdiklyje

  - Naudokite mygtukus aukštyn/žemyn/kairėn/dešinėn kameros perkėlimui pirmyn/atgal/kairėn/dešinėn
- Keiskite kameros aukštį: spauskite `Psl Aukštyn`/`Psl Žemyn`
- Keiskite kameros orientaciją (kamera išlieka savo pozicijoje, juda vaizdo centro taškas)
  - Laikykite `Ctrl` ir naudokite rodyklių mygtukus, kad pasuktumėte kamerą aukštyn, žemyn, kairėn ir dešinėn
  - Laikykite `Ctrl` ir tempkite pelę laikydami kairį mygtuką
- Priartinti ir nutolinti
  - Spauskite atitinkamai navigacijos valdiklio įrankius Priartinti ir Nutolinti
  - Sukite pelės ratuką (laikant paspaudus `Ctrl` mastelis keičiasi lėčiau)
  - Tempkite pelę paspaudus dešinį mygtuką priartinimui (tempkite žemyn) ir nutolinimui (tempkite aukštyn)

Norėdami atstatyti kameros vaizdą, spauskite mygtuką Pilnai priartinti 3D drobės skydelio viršuje.

7.2.3 Animacijos kūrimas

Animacija kuriama iš aibės kadrų - kameros pozicijų konkrečiais laiko momentais. Kad sukurtumėte animaciją:

1. Įjunkite įrankį Animacijos, bus rodomas animacijos grotuvo valdiklis
2. Spauskite mygtuką Pridėti kadrą ir įveskite *Kadro laiką* sekundėmis. Iškrentantis *Kadro* sąrašas dabar rodys ir įvestą laiką.
3. Naudodami navigacijos įrankius, perkelkite kamerą į poziciją kitam kadrui.
4. Pakartokite aukščiau išvardintus žingsnius ir pridėkite tiek kadrų (su laiku ir pozicija), kiek reikia.
5. Spauskite mygtuką animacijos peržiūrai. QGIS sukurs scenas naudojant kameros pozicijas/pasukimus nurodytais laiko momentais ir interpoliuojant papildomus kadrus tarp jų. Galima naudoti įvairius *Interpoliacijos* režimus (pvz, linijinis, inQuad, outQuad, inCirc… – daugiau informacijos rasite https://doc.qt.io/archives/qt-5.15/qeasingcurve.html#EasingFunction-typedef).

   Animaciją taipogi galima peržiūrėti tempiant laiko slankiklį. Laikant įjungtą varnelę *Ciklas*, animacija bus kartojama tol, kol bus paspaustas mygtukas .

Spauskite Eksportuoti animacijos kadrus, kad sugeneruotumėte aibę piešinių, po vieną kiekvienam kadrui. Be failo pavadinimo *Šablono* ir *Išvesties aplanko*, jūs galite nurodyti *Kadrų per sekundę* skaičių, *Išvesties plotį* ir *Išvesties aukštį*.

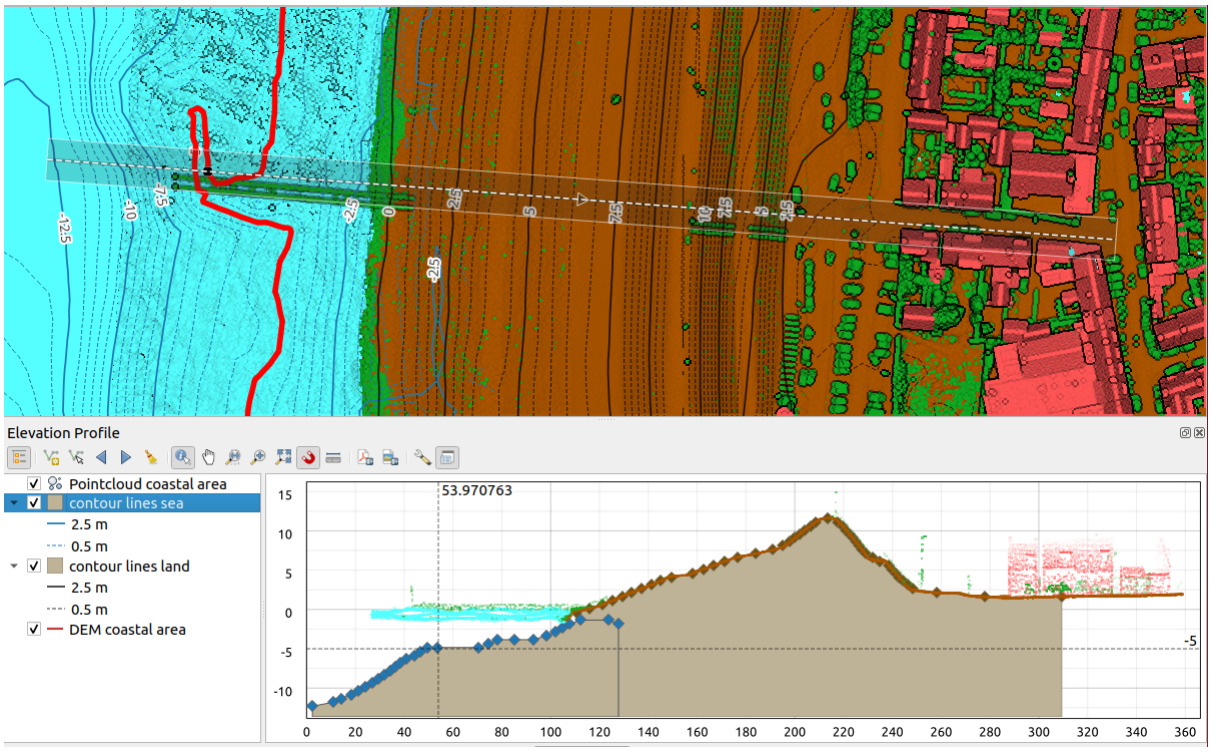
7.2.4 3D vektoriniai sluoksniai

Vektorinis sluoksnis su aukščio reikšmėmis gali būti rodomas 3D žemėlapio vaizde įjungus *Įjungti 3D braižymą* vektorinio sluoksnio savybių sekcijoje *3D vaizdas*. Keletas parinkčių leidžia valdyti vektorinio sluoksnio 3D braižymą.

7.3 Aukščio profilio vaizdas

Skydelis *Aukščio profilis* yra kraštinio vaizdo braižymo įrankis, skirtas aukščio duomenų vizualizavimui pagal liniją. Jis palaiko vektorinius, rastro, tinklelio ir taškų debesies sluoksnius. Duomenys gali būti 2D ar 3D tipo.

To add an elevation profile view, go to *View* ► *Elevation Profile* menu. You can add as many profile views as you want, and they can be docked, piled on top of each other, or floating.



7.30 Fig. : Aukščio profilio dialogas, įtrauktas žemiau pagrindinės žemėlapio drobės
Paveikslas parodo krantą su miestu už kopos. Aukščio profiliui pritaikius didesnę toleranciją grąžinama daugiau taškų debesies geoobjektų.

7.3.1 Naudotojo sąsaja

Aukščio profilio skydelio viršuje yra įrankinė su šiais įrankiais:

7.5 Table : Aukščio profilio vaizdo įrankinė

Įrankis	Kombinacija	Aprašymas
Pridėti sluoksnius	laikykite `Ctrl` tempiant ir numetant	Leidžia jums pridėti į brėžinį naujus sluoksnius
Rodyti sluoksnių medį		Rodo ar paslepia projekto sluoksnių sąrašą konfigūruojant profilio vaizdo braižymą.
Užfiksuoti kreivę		Interaktyviai virš žemėlapio drobės braižo liniją, reprezentuojančią profilio kreivę.
Užfiksuoti kreivę iš geoobjekto		Parenka esamą linijos geoobjektą žemėlapio drobėje ir sukuria profilį palei tą liniją.

continues on next page

Table 7.5 – tęsinys iš praeito puslapio

Įrankis	Kombinacija	Aprašymas
Stumtelėti kairėn	`Ctrl+Alt+,`	Leidžia jums stumtelėti fiksavimo kreivę žemėlapyje kairėn (pvz. norint rasti optimalią profilio liniją atsižvelgiant į aukštį).
Stumtelėti dešinėn	`Ctrl+Alt+.`	Leidžia jums stumtelėti fiksavimo kreivę žemėlapyje dešinėn (pvz. norint rasti optimalią profilio liniją atsižvelgiant į aukštį).
Valyti		Išima profilio liniją ir bet kokį rodomą brėžinį *Aukščio profilio* vaizde
Identifikuoti geoobjektus		Identifikuoja geoobjektus brėžinio drobėje arba per vieną paspaudimą, arba pertempiant stačiakampį. Rezultatai rodomi standartiniame skydelyje *Identifikuoti rezultatus*.
Paslinkti	`Tarpas`	Spauskite ir tempkite, kad paslinktumėte brėžinio drobę. Tai galima padaryti ir viduriniu pelės mygtuku.
Didinti X ašį		Didina/mažina pagal horizontalią ašį, išlaikant vertikalią proporciją
didinti	`Ctrl+Tarpas`	Spauskite ar spauskite ir tempkite stačiakampį virš brėžinio, kad jį padidintumėte. Laikykite `Alt` ir tada spauskite, jei norite mažinti.
Pilnai padidinti		Didina *Aukščio profilio* vaizdą pagal visą fiksacijos linijos dydį
Įjungti pritraukimą		Leidžia pritraukti iki brėžinio geoobjekto kraštinės ar viršūnės profilio vaizde. Patogi tiksliam koordinačių ar atstumų matavimui.
Matuoti atstumus		Matuoja horizontalius ir vertikalius atstumus
Eksportuoti kaip PDF		Eksportuoja brėžinį kaip PDF (kaip aukštos kokybės vektorinius objektus)
Eksportuoti kaip piešinį		Eksportuoja brėžinį į kelis piešinių formatus
Eksportuoti rezultatus		Eksportuoja brėžinius kaip 3D geoobjektus, 2D profilį ir Atstumo/Aukščio lentelę. (Rezultatus galima įrašyti kaip DXF failus, CSV failus ar į bet kokį kitą standartinį įrašomą vektorinių erdvinių duomenų formatą). Daugiau informacijos rasite skyriuje *Aukščio profilio eksportavimas*.
Parinktys ►		Teikia prieigą prie profilio aukščio linijos konfigūracijos nustatymų.
► *Užrakinti atstumo/aukščio mastelius*		Įsitikina, kad horizontalus ir vertikalus masteliai visada lygūs (taip kad pvz. 45° nuolydis profilio vaizde visada atrodys kaip 45° nuolydis).
► *Atstumo vienetai*		Leidžia profilio diagramoje braižyti atstumus kitais nei žemėlapio drobės vienetais.
► *Tolerancija*		Sets how far from the actual profile line a feature (vector point, line or polygon, point cloud) can reside to be included in the results. Uses the map units and is ignored by other layer or geometry types. For some layer types (e.g. points, lines, polygons), a custom tolerance can be set in the layer's Elevation properties tab, which replaces the global tolerance and allows more control over feature inclusion based on distance to the profile line.
► *Pervadinti profilį…*		Leidžia pervadinti profilio vaizdą

continues on next page

Table 7.5 – tęsinys iš praeito puslapio

Įrankis	Kombinacija	Aprašymas
Prijungti aukščio profilio vaizdą		Persijungia tarp prisegto ir plaukiojančio vaizdo būsenos

Paspaudus mygtuką *Rodyti sluoksnių medį* apačioje kairėje gali būti rodoma skydelio *Sluoksniai* kopija. Kaip bebūtų, tai nepriklausomas valdiklis su savo matomų sluoksnių rinkiniu bei sava rikiuote. Jis leidžia jums valdyti sluoksnių braižymą ir elgseną brėžinio drobėje:

- Uždėkite varnelę greta sluoksnio pavadinimo, kad nurodytumėte, ar reikia jį braižyti brėžinio drobėje
- Pertempkite sluoksnius aukštyn ar žemyn, kad pakeistumėte sluoksnių rikiuotę
- keiskite profilio vaizdo sluoksnių braižymo stilių: du kartus arba dešiniu mygtuku spauskite ir pažymėkite *Savybės…*, kad atvertumėte sluoksnio *Aukščio* savybių kortelę konfigūracijai. Rodoma aukščio nustatymų santrauka, kaip patarimas užvedus pelę virš sluoksnio.

Sluoksnių medžio dešinėje, brėžinio drobė yra pagrindinė vieta, kurioje galite pažiūrėti įjungtų sluoksnių aukščio profilį. Jis paremtas graduotu tinkleliu, kuriame horizontalios ašys rodo profilio linijos ilgį, o vertikalios - stebimų geoobjektų Z aukštį. Ten taipogi leidžiamos sąveikos, tokios kaip mastelio keitimas, pastūmimas, matavimas, geoobjektų identifikavimas, … naudojant viršuje esančius įrankius.

7.3.2 Aukščio profilio kūrimas

Norėdami sukurtumėte profilio vaizdą, jūs galite:

1. Eikite į meniu *Vaizdas* ► *Aukščio profilis*. Atveriamas skydelis *Aukščio profilis* panel opens.
2. Sukurkite profilio liniją, ties kuria bus braižomas paviršius ir geoobjektai. Parinkite braižymo įrankį:
   - Užfiksuoti kreivę: spauskite kairiu mygtuku pagrindinėje žemėlapio drobėje, kad pridėtumėte viršūnes, ir spauskite dešinį mygtuką, kad baigtumėte braižyti liniją, kuri bus naudojama kaip profilio linija
   - arba spauskite Užfiksuoti kreivę iš geoobjekto:, tada spauskite ant linijinio geoobjekto žemėlapio drobėje, kad pažymėtumėte jį kaip profilio liniją. Jei paspaudimo vietoje yra keli geoobjektai, apsirodys iššokantis meniu, kuriame jūs galėsite parinkti reikiamą objektą.

   Galite naudoti visas skaitmeninimo funkcijas, tokias kaip *pritraukimo parinktis*, *sekimą*, *skaitmeninimo technikas* ar išmaniojo *skaitmenizavimo skydelį*.

   Brėžinio drobė gali pradėti braižyti kai kuriuos geoobjektus.
3. Kitas žingsnis yra sukonfigūruoti aukščio savybes sluoksniams, kuriuos norite vaizduoti.

   1. Spauskite mygtuką Rodyti sluoksnių medį, kad parodytumėte sluoksnių sąrašą. Pagal nutylėjimą ne visi projekto sluoksniai įkeliami ir naudojami profilio įrankyje. Pvz. rastro sluoksniai, kurie savo savybėse nėra pažymėti kaip turintys aukščio duomenis, yra ignoruojami. Paspaudus mygtuką Pridėti sluoksnius bus rodomas filtruotas sąrašas sluoksnių, kuruos galima pridėti į brėžinį, bet kurie šuo metu nėra pridėti į brėžinį. Pritaikius dialogo pakeitimus su pažymėtais sluoksniais, jie bus automatiškai pažymėti kaip turintys aukščio duomenis ir bus nedelsiant pridėti į brėžinį.

      Patarimas

      Laikant `Ctrl` mygtuką galima tempti ir numesti papildomus sluoksnius iš pagrindinio sluoksnių medžio į aukščio profilio įrankio medį.

   2. Perjunkite jus dominančių sluoksnių matomumą. Tik jie braižomi profilio vaizde, o parinkti sluoksniai gali būti kitokie, nei pagrindiniame *Sluoksnių* skydelyje.

3. Du kartus spauskite ant sluoksnio pavadinimo ir parinkite *Savybės*. Bus atidaryta sluoksnio *Aukščio* savybių kortelė. Čia jūs konfigūruojate, kaip kiekvienas paviršiaus geoobjektas turėtų būti braižomas profilio vaizde. Galimos savybės priklauso nuo sluoksnio tipo:

   - *Rastro aukščio savybės*
   - *Vektoriaus aukščio savybės*
   - *Taškų debesies aukščio savybės*
   - *Tinklelio aukščio savybės*

   Kai tik sukonfigūruojate aukščio savybės, profilio vaizde pradedamas braižyti aktyvių sluoksnių paviršius ar geoobjektai, kertantys profilio kreivę.

4. Under Options drop-down menu, you can:

   - set the *Tolerance* value. This value is used to create a flat buffer around the elevation profile line, visible in the main map canvas. Any visible feature (point, line, polygon, etc.) overlapping that buffer will be captured in the plot canvas.
   - select the *Show Subsections Indicator* to display vertical lines at the distance of the vertices of the profile curve. This highlights the position of the vertices of the curve on the Elevation Profile.
   - select the *Subsections Symbology* to set the color, opacity, width and style of the Subsections Indicator lines.

Pastaba

Poligono išsikišimo apribojimai

Sluoksnio *Aukščio* savybėse galima nurodyti geometrijos išsikišimą, ir jis braižomas profilio vaizde. Deja, įjungus toleranciją, yra netrivialu braižyti išsikišusius poligonus, todėl, kol kas, poligonų išsikišimai ignoruojami.

7.3.3 Sąveika su profiliu vaizdais

Sukūrus profilio liniją, brėžinio drobė išdidinama iki pilnos apimties. X ašyje galite matyti profilio ilgį, o Y ašyje - aukščio diapazoną tarp minimalaus ir maksimalaus aptikto aukščio, abu žemėlapio vienetais.

Kai jūs judinate pelės žymeklį aukščio profilio vaizde, jūs pamatysite dvi susikertančias punktyrines linijas:

- vertikali linija rodo aukščio informaciją
- horizontali linija rodo atstumą nuo aukščio profilio linijos pradžios

Kai judinate pelės žymeklį aukščio profilio drobėje, jūs taipogi galite matyti ir juodą tašką, judantį palei aukščio profilio liniją pagrindinio žemėlapio drobėje. Profilio linijos viduryje, rodyklei nurodant kryptį.

Pagrindinio žemėlapio drobėje QGIS teikia galimybę naviguoti brėžinio drobėje:

- Stumti naudojamas, kai reikia pastumti aukščio profilio apimtį bet kuria jums reikiama kryptimi. Laikant mygtuką `Tarpas` pelės judinimo metu, tapogi bus pastumta ir brėžinio drobės apimtis.
- Didinti X ašį naudojamas keisti horizontalios ašies mastelį, paliekant vertikalios ašies (aukščio) proporcijas. Kairys paspaudimas ištemps brėžinį pagal X ašį, paspaustą tašką paliekant ašies centre, o stačiakampio pertempimas ištempia brėžinį pagal X ašį iki stačiakampio pločio. Laikykite `Alt` kol naudojate Didinti X ašį, jei norite keisti mastelį ties šia ašimi.
- Didinti naudojamas, kai reikia padidinti konkretų tašką (naudojant kairįjį paspaudimą) arba konkrečią apimtį (tempiant stačiakampį plotą). Laikykite `Alt` kol naudojate Didinti, jei norite atitraukti. Kartu su klavišu `Ctrl` jūs galite glotniau didinti ir mažinti mastelį.

- Didinti pilnai yra numatytasis mastelis, kuris naudojamas pradžioje ir kuris rodo pilną profilio linijos apimtį, grąžinant visus geoobjektus. Naudokite jį, kad atstatytumėte mastelio lygį.

Galima sąveikauti su brėžinio drobėje rodomais elementais:

- Spauskite mygtuką Įjungti pritraukimą, kad tiksliai pagautumėte geoobjektų taškus, viršūnes ar kraštines, tiksliam matavimui ar koordinačių ataskaitai.
- Identifikuoti geoobjektą naudojamas sluoksnių medyje matomų sluoksnių geoobjektų identifikavimui. Jūs galite tempti stačiakampį palei kelis geoobjektus profilio vaizde, jei norite tirti juos visus. Kai suderinama su formatu (pvz. vektoriai, taškų debesis), šie objektai bus paryškinti pagrindinio žemėlapio drobėje.
- Matavimų atstumai: spauskite ar pažymėkite du brėžinio drobės taškus, kad gautumėte jų horizontalų *Atstumą*, *Aukštį* ir *Suminį atstumą* tarp jų, žemėlapio vienetais.
- Stumtelėti kairėn ir Stumtelėti dešinėn naudojami, kai reikia pastumti aukščio profilio linijos poziciją žemėlapio drobėje kairėn ar dešinėn. Brėžinio drobė bus perbraižyta, rodant geoobjektus ir paviršių, dengiančius profilio linijos buferį. Linija perkeliama į šoną naudojant reikšmę *Tolerancija*, nurodyta meniu Parinktys.

Įspėjimas

Uždarius aukščio profilio vaizdą arba projektą, šiuo metu vaizdas iš projekto išimamas.

Daugiau informacijos rasite QGIS aukščio profilio/susikirtimo įrankis – detaliau!, Nyall Dawson pristatymas.

7.3.4 Aukščio profilio eksportavimas

Aukščio profilio dialogo savybių viršutinę įrankinė teikia aibę įrankių, suteikiančių galimybę eksportuoti aukščio profilį, tai leidžia jums panaudoti brėžinius kitoje programinėje įrangoje:

- Eksportuoti kaip PDF: Eksportuoja brėžinius kaip PDF (kaip aukštos kokybės vektorinius objektus). Prašoma šių parinkčių, išvesties failo kūrimui:
  - *Puslapio dydis*, parenkate puslapį iš apibrėžto sąrašo arba nurodote savo matmenis, bei puslapio orientaciją
  - *Diagramos diapazonu* valdote eksportuojamos diagramos apimtį:
    - X ašyje *Minimalus atstumas* ir *Maksimalus atstumas* nuo profilio kreivės pradžios taško
    - Y ašyje *Minimalus aukštis* ir *Maksimalus aukštis*
  - *Atstumo ašies nustatymai*: padeda jums konfigūruoti horizontalią tinklelio gradaciją virš kurios braižomas aukščio profilis. Jūs galite keisti *Užrašų intervalą*, *Pagrindinio tinklelio intervalą* ir *Antrinio tinklelio intervalą*.
  - *Aukščio ašies nustatymai*, tas pats, kaip aukščiau, tik vertikaliam gradavimui
- Eksportuoti kaip piešinį: Eksportuoja brėžinius į piešinių formato failą. Paprašoma įvesti konkrečias *Eksportavimo parinktis*, tokias kaip *Piešinio plotį* ir *Piešinio aukštį*, kartu su *Brėžinio intervalus*, *Atstumo ašių nustatymus* ir anksčiau minėtus *Aukščio ašies nustatymus*.
- Eksportuoti rezultatus: Eksportuoti profilio rezultatus kaip aibę vektorinio sluoksnio geoobjektų (`.DXF`, `.CSV`, `.SHP`, `.GPKG`, ...). Priklausomai nuo išvesties formato, gali būti grąžinti keli sluoksniai, kai eksportuojamų geoobjektų geometrijų tipai skiriasi.
  - *Eksportuoti 3D geoobjektus*: Eksportuoja profilio linijas kaip 3D geoobjektus, aukščio reikšmės imamos iš aukščio pjūvių ir laikomos eksportuotų geometrijų Z reikšmėse.

- *Eksportuoti 2D profilį*: Eksportuoja profilį kaip 2D profilio linijas, kur aukštis saugomas eksportuojamos geometrijos Y matmenyje, o atstumas - X matmenyje (kaip rodoma aukščio profilio valdiklyje).
- *Eksportuoti atstumo/aukščio lentelę*: Eksportuoja profilį kaip lentelę su atstumo ir aukščio reikšmėmis.

SKYRIUS 8

Bendri įrankiai

8.1 Kontekstinė pagalba

Kai jums reikia pagalbos konkrečia tema, jūs galite žiūrėti atitinkamą Naudotojo vadovo puslapį per mygtuką *Pagalba*, kurį rasite daugumoje dialogų — prašome pastebėti, kad trečiųjų šalių priedai gali nukreipti jus į savo svetainių puslapius.

8.2 Skydeliai

Pagal nutylėjimą QGIS darbui teikia daug skydelių. Kai kurie iš jų aprašomi žemiau, o kitus galima rasti kitose dokumentacijos dalyse. Pilną QGIS teikiamų numatytųjų skydelių sąrašą galima rasti meniu *Vaizdas* ► *Skydeliai* ► ir rasti juos paminėtus skyriuje *Skydeliai*.

8.2.1 Sluoksnių skydelis

Skydelyje *Sluoksniai* (dar žinomame kaip `žemėlapio legenda`) yra visų projekto sluoksnių sąrašas ir jis padeda jums valdyti jų matomumą bei žemėlapio išvaizdą. Jūs galite įjungti ir išjungti šį skydelį spausdami `Ctrl+1`.

QGIS suteikia daugybę būdų pridėti sluoksnius į projektą:

- naudojant mygtuką *Pridėti* iš konkretaus duomenų tiekėjo kortelės dialoge *Duomenų šaltinių tvarkyklė*
- QGIS skydelyje *Naršyklė* ar *DB tvarkyklė*: spauskite du kartus, tempkite ir numeskite failus ir sluoksnius į QGIS arba naudokite kontekstinį meniu
- tempkite ir numeskite failus iš operacinės sistemos failų naršyklės į QGIS

Visais šiais atvejais jūs galite atverti vieną ar daugiau sluoksnių vienu metu. Nauji sluoksniai pridedami į skydelį *Sluoksniai*:

1. jei numetami virš skydelio *Sluoksniai* - toje konkrečioje vietoje, kur jie paleidžiami
2. kitais atvejais vietoje, kuri nurodyta globaliame nustatyme *Elgsena pridedant naujus sluoksnius*
3. o pridedant kelis sluoksnius jie rikiuojami tokia tvarka, kuri padidina galimybę, kad jų grupavimas logiškas ir geoobjektai kiek galima geriau matomi, naudojant tokią logiką (nuo viršaus žemyn):
    - vektorinių taškų sluoksniai

- vektorinių linijų sluoksniai
- vektorinių poligonų sluoksniai
- taškų masyvų sluoksniai
- tinklelio sluoksniai
- rastro sluoksniai

Įrankinė *Sluoksnių* skydelio viršuje leidžia jums:

- Atverti sluoksnių stilių skydelį (F7): įjungti ir išjungti *Sluoksnių stiliaus* skydelį.
- Pridėti naują grupę: žr. *Sąveika su grupėmis ir sluoksniais*
- Valdyti žemėlapio temas: valdyti sluoksnių matomumą ir tvarkyti juos skirtingose *žemėlapio temose*.
- filtruoti sluoksnius legendos medyje:
  - *Filtruoti legendą pagal žemėlapio turinį*: sluoksnių skydelyje rodomi tik sluoksniai, kurie yra nurodyti kaip matomi, ir kurių geoobjektai kerta dabartinę žemėlapio drobę. Priešingu atveju sluoksniui taikomas bendras NULL simbolis. Priklausomai nuo sluoksnio simbologijos tai yra patogus būdas identifikuoti, kurio sluoksnio geoobjektai yra jūsų dominančioje teritorijoje.
  - *Rodyti privačius sluoksnius*: patogus būdas rodyti ir sąveikauti su *privačiais sluoksniais* skydelyje *Sluoksniai* nekeičiant projekto nustatymų.
  - *Rodyti tik neveikiančius sluoksnius*: rodomi tik sluoksniai su neveikiančiais duomenų šaltiniais.
- Filtruoti legendą pagal išraišką: pritaikyti išraišką, kad iš parinktų sluoksnių medžio būtų išimti stiliai, kurie neturi geoobjektų, tenkinančių sąlygą. Tai galima naudoti paryškinti geoobjektus, kurie yra kito sluoksnio ploto/geoobjekto viduje. Iškrentančiame sąraše jūs galite keisti ar išvalyti šiuo metu taikomą išraišką.
- Išskleisti viską ar Sutraukti viską - išskleidžia ar sutraukia sluoksnius ir grupes sluoksnių skydelyje.
- Išimti sluoksnių grupę, kuri šiuo metu pažymėta.

8.1 Fig. : Sluoksnių skydelio sluoksnių įrankinė

Pastaba

Įrankiai sluoksnių skydelio valdymui taipogi veikia ir žemėlapiui bei legendos elementams spausdinimo išdėstymuose

Žemėlapio temų konfigūravimas

Iškrentantis mygtukas Tvarkyti žemėlapio temas leidžia greitai ir patogiai valdyti sluoksnių matomumą skydelyje *Sluoksniai*:

- *Rodyti visus sluoksnius*
- *Slėpti visus sluoksnius*
- *Rodyti parinktus sluoksnius*

- *Slėpti parinktus sluoksnius*
- *Perjungti parinktus sluoksnius*: keičia pirmo skydelyje parinkto sluoksnio matomumą ir taiko tą būseną visiems kitiems parinktiems sluoksniams. Taipogi pasiekiamas naudojant klavišą `Tarpas`.
- *Perjungti parinktus sluoksnius nepriklausomai*: keičia kiekvieno parinkto sluoksnio matomumo būseną
- *Slėpti neparinktus sluoksnius*

Be paprasto sluoksnių matomumo valdymo, meniu Tvarkyti žemėlapio temas leidžia jums konfigūruoti **Žemėlapio temas** legendoje ir persijungti iš vienos žemėlapio temos į kitą. Žemėlapio tema yra dabartinės žemėlapių legendos **nuotrauka**, kuri atsimena:

- sluoksnius, kurie pažymėti kaip matomi skydelyje *Sluoksniai*
- **ir** kiekvienam matomam sluoksniui:
  - nuorodą į *stilių*, kuris pritaikytas kiekvienam sluoksniui
  - matomos stiliaus klasės, t.y. *Sluoksnio skydelyje* įjungti elementai. Tai taikoma *simbologijoms*, kitokioms nei vieno simbolio braižymas
  - sluoksnio elemento(ų) ir vidinės grupės(ių) suskleidimo/išskleidimo būsena

Norėdami sukurti žemėlapio temą:

1. Įjunkite sluoksnį, kurį norite rodyti
2. Įprastai sukonfigūruokite sluoksnio savybes (simbologiją, diagramas, užrašus…)
3. Išplėskite meniu apačioje *Stilius* ► ir spauskite *Pridėti…*, kad įrašytumėte nustatymus kaip *naują projekte įdėtą stilių*

   > **Pastaba**
   >
   > Žemėlapio tema neatsimena dabartinių savybių detalių: įrašoma tik nuoroda į stiliaus pavadinimą, taigi kai keičiate įjungtą sluoksnį (pvz. keičiate simbologijos braižymą), informacija atnaujinama žemėlapio temoje.

4. Kartokite ankstesnius žingsnius tiek kiek reikia kitiems sluoksniams
5. Jei reikia, išplėskite ar suskleiskite grupes ar matomų sluoksnių elementus skydelyje *Sluoksniai*
6. Spauskite mygtuką Tvarkyti žemėlapio temas skydelio viršuje ir tada *Pridėti temą…*
7. Įveskite žemėlapio temos pavadinimą ir spauskite *Gerai*

Nauja tema rodoma apatinėje iškrentančio meniu dalyje.

Galite kurti tiek žemėlapio temų, kiek reikia: kai tik dabartinė žemėlapio legendos kombinacija (matomi sluoksniai, jų aktyvumo būsena, žemėlapio legendos elementai) neatitinka esamos temos turinio, kaip nurodyta viršuje, spauskite *Pridėti temą…*, kad sukurtumėte naują žemėlapio temą, arba naudokite *Keisti temą* ►, kad pakeistumėte esamą žemėlapio temą. Galite pervadinti aktyvią žemėlapio temą naudodami *Pervadinti dabartinę temą…* arba naudokite mygtuką *Išimti dabartinę temą*, jei norite ją ištrinti.

Žemėlapio temos padeda greitai persijungti tarp skirtingų iš anksto sukonfigūruotų kombinacijų: parinkite žemėlapio temą sąraše, kad atstatytumėte jos kombinaciją. Visos sukonfigūruotos temos taipogi prieinamos ir spausdinimo išdėstymuose, kur jūs galite sukurti skirtingus žemėlapio elementus pagal konkrečias temas, nepriklausomai nuo to, kas braižoma dabartinėje pagrindinėje drobėje (žr. *Žemėlapio elementų sluoksniai*).

Sluoksnių skydelio kontekstinio meniu apžvalga

Po įrankine yra pagrindinis Sluoksnių skydelio komponentas - sąrašas visų į projektą pridėtų sluoksnių, pasirinktinai sudėliotas į grupes. Sluoksnis su įjungta greta jo esančia varnele rodo savo turinį žemėlapio drobės apimtyje, nebent nustatytas *nuo mastelio priklausantis matomumas*. Sluoksnį galima pažymėti ir tempti aukštyn ar žemyn legendoje, norint pakeisti jo Z-rikiuotę. Z-rikiuotė reiškia, kad aukščiau išvardinti sluoksniai braižomi virš sluoksnių, išvardintu legendos apačioje. Taipogi sluoksnį ar sluoksnių grupę galima tempti iš vieno QGIS egzemplioriaus į kitą.

> **Pastaba**
>
> Z-rikiuotės elgseną galima keisti skydelyje *Sluoksnių rikiuotė*.

Depending on the item selected in the panel, a right-click shows a dedicated set of options presented below. Some of these entries are not available when multiple selections are made (e.g., *Filter*, *Rename*, *Properties*).

8.1 Table : *Sluoksnių* skydelio kontekstinio meniu elementai

Parinktis	Grupė	Vektorinis sluoksnis	Rastro sluoksnis	Tinklelio sluoksnis	Taškų masyvo sluoksnis	3D sluoksnis
Didinti iki sluoksnio(ių)/grupės	✓	✓	✓	✓	✓	✓
Didinti iki pažymėjimo		✓				
Rodyti apžvalgoje		✓	✓	✓	✓	✓
Rodyti objektų skaičių		✓				
Rodyti užrašą		✓		✓		
Kopijuoti sluoksnį/grupę	✓	✓	✓	✓	✓	✓
Pervadinti sluoksnį/grupę	✓	✓	✓	✓	✓	✓
Didinti iki savos rezoliucijos (100%)			✓			
Tempti naudojant dabartinę apimtį			✓			
Create Raster Attribute Table			✓			
Open Raster Attribute Table			✓			
Load Raster Attribute Table from VAT.DBF			✓			
Atnaujinti SQL sluoksnį…		✓				
Execute SQL…		✓				
Keisti virtualų sluoksnį…		✓				
Pridėti grupę	✓					
Dubliuoti sluoksnį		✓	✓	✓	✓	✓
Išimti sluoksnį/grupę…	✓	✓	✓	✓	✓	✓
Išimti iš grupės		✓	✓	✓	✓	
Perkelti į viršų	✓	✓	✓	✓	✓	
Perkelti į apačią	✓	✓	✓	✓	✓	
Įjungti su visais tėvais		✓	✓	✓	✓	
Grupuoti parinktus		✓	✓	✓	✓	

continues on next page

Table 8.1 – tęsinys iš praeito puslapio

Parinktis	Grupė	Vektorinis sluoksnis	Rastro sluoksnis	Tinklelio sluoksnis	Taškų masyvo sluoksnis	3D sluoksnis
Atverti atributų lentelę		☑				
Perjungti redagavimą		☑		☑		
Dabartiniai keitimai ►		☑		☑		
Filtras…		☑	☑		☑	
Keisti duomenų šaltinį…		☑	☑	☑	☑	
Taisyti duomenų šaltinį…		☑	☑	☑	☑	
Veiksmai su pažymėjimais ► (redagavimo režime)		☑				
► *Dubliuoti geoobjektą*		☑				
► *Dubliuoti geoobjektą ir skaitmeninti*		☑				
Nustatyti sluoksnio mastelio matomumą…		☑	☑	☑	☑	☑
Didinti iki matomo mastelio		☑	☑	☑	☑	
Sluoksnio CRS ►		☑	☑	☑	☑	☑
► *Nustatyti projekto CRS iš sluoksnio*		☑	☑	☑	☑	☑
► *Nustatyti į..* (paskutiniai CRS)				☑	☑	☑
► *Nustatyti sluoksnio CRS…*		☑	☑	☑	☑	☑
Nustatyti grupės CRS…	☑					
Nustatyti grupės WMS duomenis…	☑					
Viena kitą išjungianti grupė	☑					
Įjungti su visais vaikais (Ctrl-paspaudimas)	☑					
Išjungti su visais vaikais (Ctrl-paspaudimas)	☑					
Daryti pastoviu		☑				
Eksportuoti ►	☑	☑	☑	☑	☑	☑
► *Įrašyti kaip…*			☑			
► *Įrašyti geoobjektus kaip…*		☑				
► *Įrašyti parinktus geoobjektus kaip…*		☑				
► *Įrašyti kaip sluoksnio apibrėžimo failą…*	☑	☑	☑	☑	☑	☑
► *Įrašyti kaip QGIS sluoksnio stiliaus failą…*		☑	☑	☑	☑	☑
Stiliai ►		☑	☑	☑	☑	☑
► *Kopijuoti stilių*		☑	☑	☑	☑	☑
► *Įkelti stilių*	☑	☑	☑	☑	☑	
► *Pridėti…*		☑	☑	☑	☑	☑
► *Pervadinti dabartinį…*		☑	☑	☑	☑	☑

continues on next page

Table 8.1 – tęsinys iš praeito puslapio

Parinktis	Grupė	Vektorinis sluoksnis	Rastro sluoksnis	Tinklelio sluoksnis	Taškų masyvo sluoksnis	3D sluoksnis
► *Keisti simbolį…*		☑				
► *Kopijuoti simbolį*		☑				
► *Įkelti simbolį*		☑				
Pridėti sluoksnio pastabas…		☑	☑	☑	☑	☑
Keisti sluoksnio pastabas…		☑	☑	☑	☑	
Išimti sluoksnio pastabas		☑	☑	☑	☑	
Savybės…		☑	☑	☑	☑	☑

GRASS vektoriniams sluoksniams negalimas Perjungti redagavimą. Daugiau informacijos apie GRASS vektorinių sluoksnių redagavimą rasite skyriuje *Digitizing and editing a GRASS vector layer*.

Sąveika su grupėmis ir sluoksniais

Sluoksnius legendų lange galima organizuoti į grupes. Yra keli būdai, kaip tai galima padaryti:

1. Spauskite piktogramą, kad pridėtumėte naują grupę. Įrašykite grupės pavadinimą ir spauskite `Enter`. Dabar spauskite esamą sluoksnį ir įtempkite jį į grupę.
2. Parinkite daugiau nei vieną sluoksnį, spauskite piktogramą . Parinkti sluoksniai automatiškai įterpiami į naują grupę.
3. Parinkite kelis sluoksnius, spauskite dešinį pelės mygtuką legendos lange ir parinkite *Grupuoti parinktus*. Parinkti sluoksniai bus automatiškai įdėti į naują grupę.

Jei norite iškelti sluoksnį iš grupės, ištempkite jį arba spauskite ant jo dešinį mygtuką ir pasirinkite *Iškelti iš grupės*: sluoksnis bus iškeltas iš grupės ir padėtas virš jos. Grupes taipogi galima įdėti į kitas grupes. Jei sluoksnis yra tokioje kelių lygmenų grupėje, *Iškelti iš grupės* iškels sluoksnį iš visų grupių.

Jei norite perkelti grupę ar sluoksnį į sluoksnių skydelio viršų, arba pertempkite jį į viršų, arba pasirinkite *Perkelti aukštyn*. Jei šią parinktį naudosite sluoksniui grupėje, sluoksnis bus perkeltas į dabartinės grupės viršų. Parinktis *Perkelti žemyn* turi tą pačią logiką, tik perkelia sluoksnius ir grupes žemyn.

Grupės varnelė vienu paspaudimu parodys arba paslėps grupėje esančius sluoksnius. Laikant `Ctrl` varnelė taipogi įjungs ar išjungs visus sluoksnius grupėje ir pogrupiuose.

`Ctrl`-paspaudimas ant įjungto / išjungto sluoksnio išjungs / įjungs sluoksnį ir visus jo tėvus.

Įjungus parinktį **Viena kitą išjungianti grupė**, jūs grupėje vienu metu galite įjungti tik vieną sluoksnį. Kai tik tokios grupės sluoksnis įjungiamas, visi kiti sluoksniai yra išjungiami.

Galima tuo pačiu metu parinkti daugiau nei vieną sluoksnį ar grupę laikant mygtuką `Ctrl` ir spaudžiant ant papildomų sluoksnių. Jūs tada galite perkelti visus parinktus sluoksnius į naują grupę vienu metu.

Jūs taipogi galite vienu metu ištrinti daugiau nei vieną sluoksnį ar grupę, parenkant kelis elementus su `Ctrl` klavišu ir tada spaudžiant `Ctrl+D`: visi parinkti sluoksniai ar grupės bus išimti iš sluoksnių sąrašo.

Daugiau informacijos apie sluoksnius ir grupes naudojant indikatoriaus piktogramą

Kai kuriais atvejais, sluoksnių ar grupių formatavimas ar greta skydelyje *Sluoksniai* esančios piktogramos duoda daugiau informacijos apie sluoksnį/grupę. Šie elementai yra:

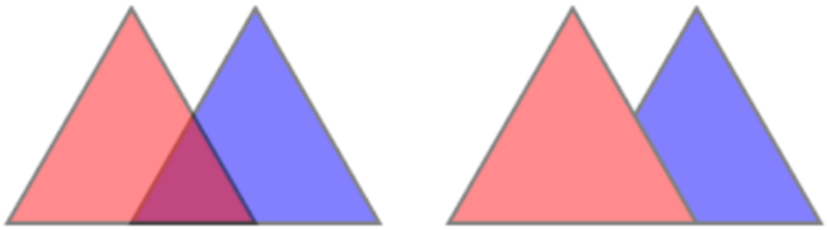
- rodo, kad sluoksnis yra redagavimo režime ir jūs galite keisti duomenis
- rodo, kad sluoksnis yra redaguojamas ir yra neįrašytų pakeitimų
- rodo, kad sluoksniui taikomas *filtras*. Užveskite pelę virš piktogramos, kad pamatytumėte filtro išraišką, spauskite du kartus, jei norite pakeisti užklausą
- rodo, kad sluoksniai šiame projekte yra *privalomi*, todėl jų negalima išimti
- identifikuoja *įdėtą grupę ar sluoksnį* ir kelią iki jų originalaus projekto failo
- identifikuoja sluoksnį, kurio duomenų šaltinis buvo neprieinamas projekto atidarymo metu (žr. *Neteisingų kelių iki failų tvarkymas*). Spauskite piktogramą, kad atnaujintumėte šaltinio kelią arba sluoksnio kontekstiniame meniu parinkite *Taisyti duomenų šaltinį…*.
- primena jums, kad sluoksnis yra *laikinas juodraštinis sluoksnis* ir jo turinys bus išmestas, kai jūs uždarysite šį projektą. Kad išvengtumėte duomenų praradimo, paverskite sluoksnį pastoviu paspausdami piktogramą, kad įrašytumėte sluoksnį į bet kokį GDAL vektorinį formatą, palaikomą QGIS.
- identifikuoja sluoksnį, kuris naudojamas *redagavimo režime neprisijungus*.
- identifikuoja sluoksnį, kuris neturi ar turi nežinomą CRS
- sluoksniai su koordinatėmis, laikomomis koordinačių atskaitos sistemoje, kuri iš principo turi žemą tikslumą (reikia, kad būtų įjungta *atitinkama nuostata*)
- identifikuoja drobės animacijos valdomą sluoksnį su laiku
- identifikuoja sluoksnį, kuris turi susijusias *pastabas*
- Pilkas pavadinimas, kai žemėlapio drobės mastelis nepatenka į sluoksnio matomų mastelių diapazoną (nustatoma *Braižymo* savybėse). Spauskite kontekstinio meniu parinktį *Didinti iki matomo mastelio*, kad pakeistumėte žemėlapio mastelį iki artimiausio mastelio, kuriame sluoksnis bus matomas.

Sluoksnių braižymo kontrolė naudojant grupavimą

Grupės - tai būdas struktūrizuoti sluoksnius projekto medyje, bet jos taipogi gali įtakoti kaip braižomas jų turinys - kaip vienas objektas braižymo metu.

Tokio braižymo parinktis galima skydelyje *Sluoksnio stilius*, kai parinkta grupė. Kortelėje Simbologija įjunkite *Braižyti sluoksnius kaip grupę*, kad įjungtumėte keletą parinkčių, kurios valdo bendrą vaikinių sluoksnių išvaizdą, vietoj atskirų sluoksnių:

- *Nepermatomumas*: Vaikinių sluoksnių geoobjektai, kuriuos dengia kitas vaikinis sluoksnis liks uždengti, nepermatomumas taikomas tik „visai grupei“.



8.2 Fig. : Nepermatomumas nurodytas sluoksniams ir grupei
Piešinukas kairėje rodo du sluoksnius, kurie braižomi su 50% nepermatomumu (matomi ir žemiau esantys geoobjektai, bet pusiau uždengti 50% raudono geoobjekto viršuje). Antrame paveiksliuke matomas rezultatas, kai nepermatomumas nurodytas grupei (žemiau esančio mėlyno geoobjekto dalys yra pilnai uždengtos virš jo esančio raudono sluoksnio, o gautas rezultatas braižomas su 50% nepermatomumu).

- *Suliejimo režimai*: Kaip ir nepermatomumas, *suliejimo režimo* nustatymas (pvz. daugybos, perdengimo, …) visai grupei reiškia, kad iš pradžių paruošiami vaikiniai sluoksniai, kur aukščiau esantys uždengia žemesniuosius. Braižymas tada gaunamas suliejant plokščią grupę su sluoksniais, esančiais žemiau grupės.
  - Kai vaikiniams sluoksniams priskirti suliejimo režimai, jis taikomas prieš išplokštinimą, bet apimtis ribojama tik tos grupės vaikiniams sluoksniams, o ne kitiems sluoksniams, esantiems žemiau visos grupės.
  - Grupių vaikiniams sluoksniams pateikiama daugiau *suliejimo režimų* jų kortelėje *Simbologija*, kurie atlieka stiliaus „apkarpymo“ veiksmus kitiems vaikiniams sluoksniams braižymo metu. Pavyzdžiui jūs galite apkarpyti vieno sluoksnio turinio braižymą turiniu iš kito „uždengiančio“ sluoksnio.
- *Sluoksnio efektai*: *efektai* taikomi tik suplokštintam vaikinių sluoksnių vaizdui, taigi pavyzdžiui metamo šešėlio efektas bus taikomas uždengtiems vaikiniams sluoksniams.

Kai grupei nurodoma *Braižyti sluoksnius kaip grupę*, tada tik grupė bus rodoma skydelyje *Sluoksnių rikiuotė*. Grupių vaikai nebus matomi šiame rikiuotės sąraše, nes jų rikiuotę nustato grupės sluoksnio vieta.

Sluoksnio stiliaus keitimas

Skydelyje *Sluoksniai* jūs turite kelis variantus, kaip greitai ir paprastai pakeisti sluoksnio braižymą.

Spauskite ant sluoksnio dešinį pelės mygtuką ir parinkite *Stiliai* ►, kad:

- pamatytumėte šiuo metu sluoksniui galimus *stilius*. Jei apibrėžėte daug stilių, galite perjungti iš vieno į kitą ir jūsų sluoksnio braižymas bus automatiškai atnaujintas žemėlapio drobėje.
- kopijuotumėte dalį ar visą dabartinį stilių, o, kai galima, įkeltumėte nukopijuotą stilių iš kito sluoksnio
- *Pervadinti dabartinį…* stilių
- *Pridėti* naują stilių (kuris yra dabartinio sluoksnio kopija)
- arba *Pašalinti dabartinį* stilių (tik kai jau yra keli stiliai).

> **Patarimas**
>
> **Greitai dalinkitės sluoksnio stiliumi**
>
> Kontekstiniame meniu nukopijuokite sluoksnio stilių ir įkelkite jį į grupę ar sluoksnių pažymėjimą: stilius taikomas visiems to paties tipo (vektoriniams, rastro, tinklelio, taškų masyvo, …) sluoksniams, kaip pradinis sluoksnis ir, vektoriniams sluoksniams, turi tą patį geometrijos tipą (taškas, linija ar poligonas).

Kai naudojama simbologija, kuri remiasi geoobjekto klasifikacija (pvz. *kategorizuota*, *graduota* ar *pagal taisykles* vektoriniams sluoksniams ar *klasifikuota* taškų masyvams), paspaudus dešinį pelės mygtuką ant klasės įrašo skydelyje *Sluoksniai* galima pakeisti klasių (ir jų geoobjektų) matomumą ir išvengti jų (ne)pažymėjimo po vieną:

- *Perjungti elementus*
- *Rodyti visus elementus*
- *Slėpti visus elementus*

Vektoriniams sluoksniams, klasės šakos įrašo kontekstinis meniu taipogi leidžia:

- *Pažymėti geoobjektus*: pažymi visus klasę atitinkančius sluoksnio geoobjektus
- *Rodyti atributų lentelėje*: atidaro atributų lentelę, kurioje filtruoti tik atitinkančios klasės geoobjektai
- keisti *simbolio spalvą* naudojant **Spalvos ratą**. Patogumui, spalvų rato apačioje taipogi galima pasirinkti ir iš paskutinių naudotų spalvų.
- *Keisti simbolį…*: atveria dialogą *Simbolio parinkimas*, leidžiantį keisti geoobjekto simbolį (simbolį, dydį, spalvą…).
- *Kopijuoti simbolį*
- *Įkelti simbolį*

Patarimas

Du kartus paspaudus klasės lapo įrašą taipogi atveriamas dialogas *Simbolio parinkimas*.

8.2.2 Sluoksnio stiliaus skydelis

Skydelis *Sluoksnio stilius* (dar įjungiamas su `Ctrl+3`) yra greitas kelias iki to paties funkcionalumo, kaip ir randamam dialoge *Sluoksnio savybės*. Jis suteikia greitą ir paprastą būdą apibrėžti sluoksnio braižymą ir elgseną bei vizualizuoti rezultatus neatidarant sluoksnio savybių dialogo.

Be to, kad vengia sluoksnio savybių dialogo blokavimo (arba „modalizavimo“), sluoksnio stiliaus skydelis taip pat išvengia ekrano užgriozdinimo dialogais ir turi daugumą stiliaus funkcijų (spalvų parinkiklį, efektų savybes, taisyklių keitimą, užrašų pakeitimą…): pvz., paspaudus spalvų mygtukus sluoksnio stiliaus skydelių viduje, spalvos parinkimo dialogas atidaromas sluoksnio stiliaus skydelio viduje, o ne kaip atskiras dialogas.

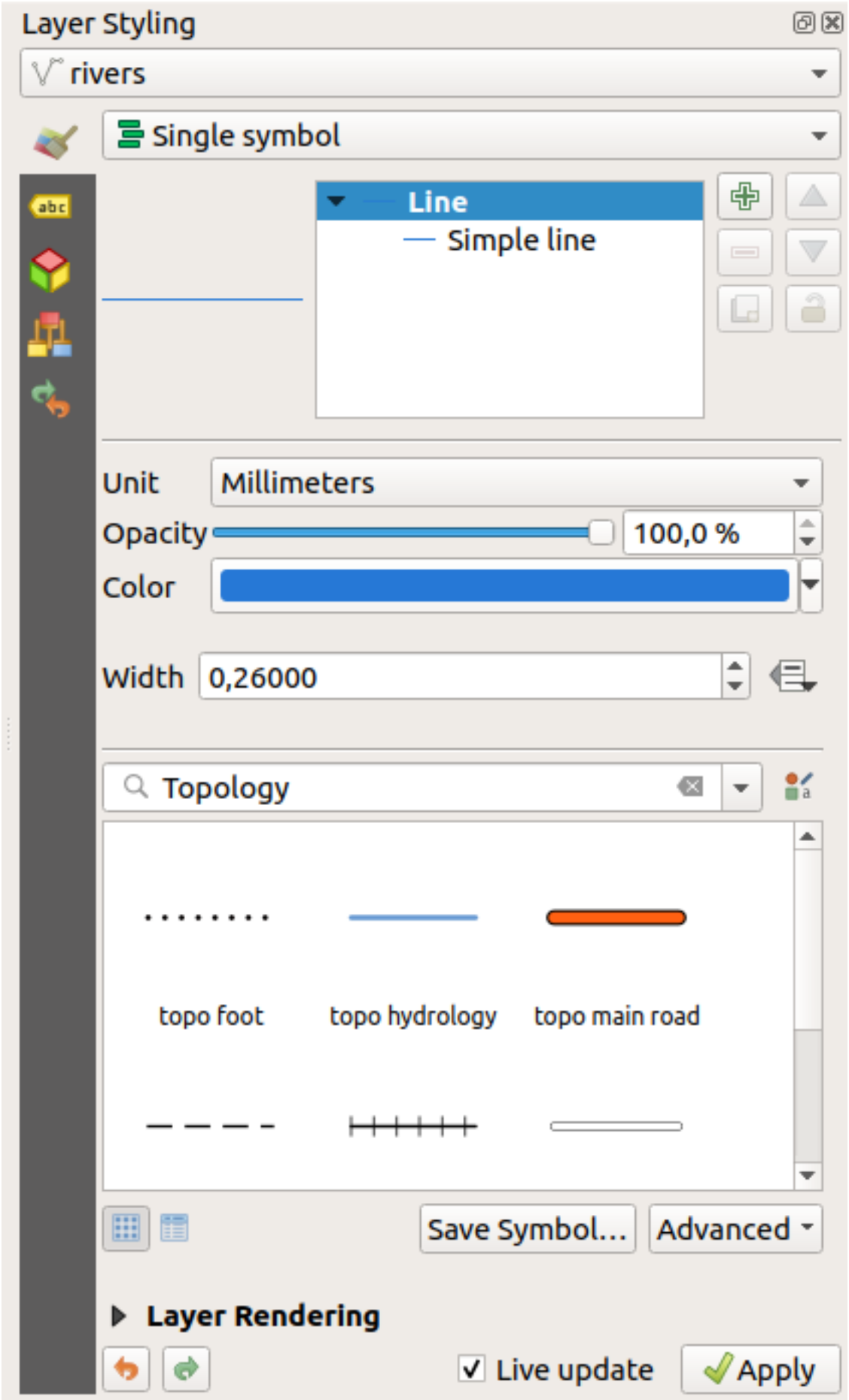
Sluoksnio skydelyje esančiame iškrentančiame dabartinių sluoksnių sąraše parinkite elementą ir:

- Priklausomai nuo aktyvaus elemento nustatykite:
  - Grupių *simbologiją* (žr. *Sluoksnių braižymo kontrolė naudojant grupavimą*)
  - Rastro sluoksnio *simbologijos*, *permatomumo* ir *histogramos* savybes. Šios parinktys tokios pačios kaip ir *Rastro savybių dialogas*.
  - Vektorinių sluoksnių *simbologijos*, *užrašų*, *slėpimo* ir *3D vaizdo* savybes. Šios parinktys yra tokios pačios kaip ir *Vektorių savybių dialogas* ir gali būti išplėstos savo savybėmis, kurias prideda trečiųjų šalių priedai.
  - *Symbology*, *Labels* and *3D View* properties for mesh layer. These options are the same as in the *Mesh Dataset Properties*.
  - *Symbology*, *3D View* and *Elevation* properties for point cloud or 3D Tiles layer. These options are the same as in the *Point Clouds Properties* or *3D Tiles Properties*.
  - *Symbology* and *Labels* properties for vector tiles layer. These options are the same as in the *Vector Tiles Dataset Properties*.

  - *Symbology* properties for annotations layer. These options are the same as in the *feature annotation properties*.

- Įjunkite ir konfigūruokite *globalias žemėlapio šešėliavimo* savybes
- Tvarkykite susijusius stilius *Stilių tvarkyklėje* (daugiau informacijos *Savo stilių tvarkymas*).
- Žiūrėkite jūsų padarytų dabartinio projekto sluoksnio pakeitimų *Istoriją*, kur galite ir atšaukti ar atstatyti bet kokią būseną parinkdami ją ir spausdami *Taikyti*.

Kita galinga šio skydelio savybė yra varnelė *Gyvas atnaujinimas*. Įjunkite ją, jei norite, kad jūsų pakeitimai nedelsiant būtų atspindėti žemėlapio drobėje: jums daugiau nebereikės spaudyti mygtuko *Taikyti*.



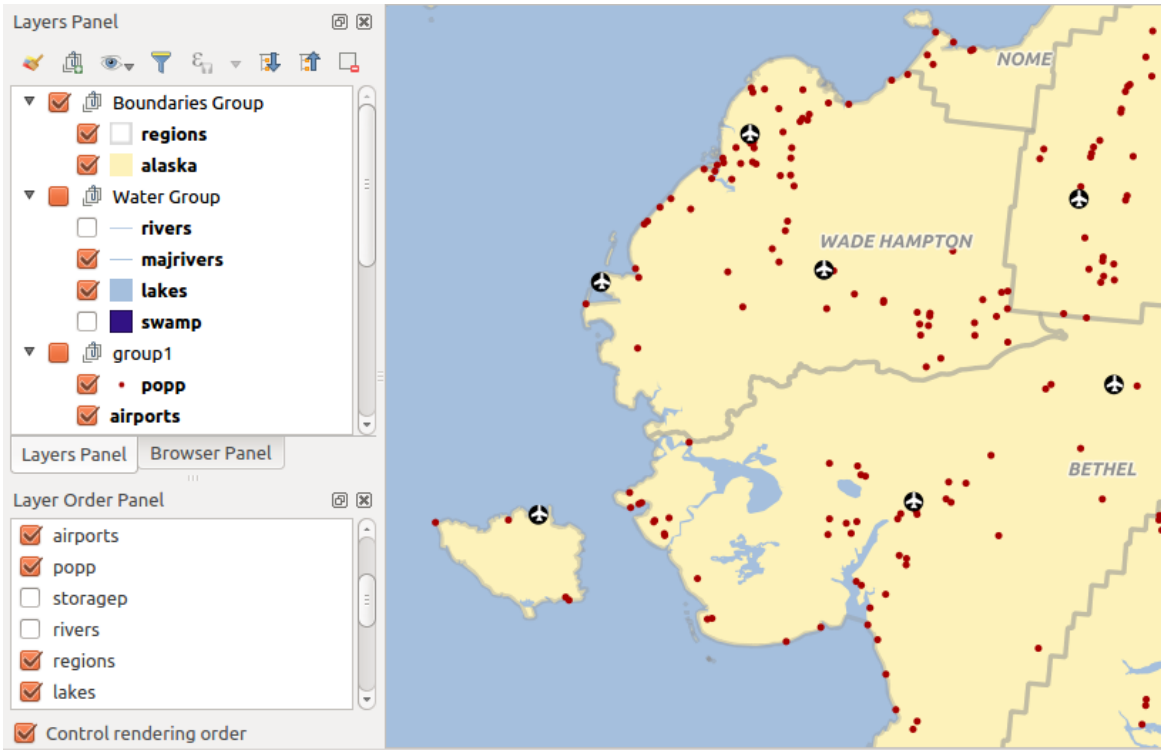
8.3 Fig. : Sluoksnio simbologijos nustatymas stilių skydelyje

8.2.3 Sluoksnių rikiuotės skydelis

Pagal nutylėjimą QGIS žemėlapio drobėje sluoksniai rodomi pagal skydelyje *Sluoksniai* matomą seką: kuo aukščiau šiame skydelyje yra sluoksnis, tuo aukščiau (taigi matomiau) sluoksnis bus ir žemėlapio vaizde.

Galite nustatyti sluoksnių braižymo tvarką, kuri nepriklausytų nuo sluoksnių skydelio, naudodami skydelį *Sluoksnių eiliškumas*, kurį galite įjungti meniu *Vaizdas* ► *Skydeliai* ► arba kombinacija `Ctrl+9`. Įjunkite varnelę *Valdyti braižymo eiliškumą*, esančia sąrašo apačioje, kad surikiuotumėte sluoksnius taip, kaip jums reikia. Ši rikiuotė taikoma žemėlapio drobei. Pavyzdžiui Fig. 8.4, jūs matote, kaip geoobjektai `oro uostai` yra rodomi virš `alaska` poligono, nepriklausomai nuo to, kaip tie sluoksniai išdėstyti sluoksnių skydelyje.

Išjungus varnelę *Valdyti braižymo eiliškumą* bus atstatyta numatytoji elgsena.



8.4 Fig. : Nuo legendos nepriklausančio sluoksnių eiliškumo nustatymas

8.2.4 Apžvalgos skydelis

Skydelis *Peržiūra* (`Ctrl+8`) rodo žemėlapį su kai kurių sluoksnių pilna apimtimi. Peržiūros žemėlapis užpildomas sluoksniais naudojant meniu *Sluoksniai* ar sluoksnių kontekstiniame meniu esančią parinktį *Rodyti peržiūroje*. Šiame vaizde raudonas stačiakampis rodo dabartinės žemėlapio drobės apimtį, taip padedant jums greitai susiorientuoti, kurią žemėlapio vietą šiuo metu žiūrite. Jei paspausite ir tempsite raudoną stačiakampį peržiūros ribose, atitinkamai keisis ir pagrindinio žemėlapio vaizdo apimtis.

Pastebėtina, kad peržiūros žemėlapyje nebraižomi užrašai, net jei žemėlapyje sluoksniams yra sukonfigūruotas užrašų rodymas.

8.2.5 Pranešimų žurnalo skydelis

Įkeliant ar vykdant kai kuriuos apdorojimo veiksmus jūs galite sekti pranešimus, kurie pasirodo skirtingose *Pranešimų žurnalo skydelio* kortelėse. Jį galima įjungti naudojant patį dešiniausią būsenos juostos mygtuką.

8.2.6 Atšaukimo/pakartojimo skydelis

Kiekvienam keičiamam sluoksniui skydelis *Atšaukimas/pakartojimas* (`Ctrl+5`) rodo visų įvykdytų veiksmų sąrašą, leidžia jums greitai atšaukti kelis veiksmus parenkant aukščiau pateiktą veiksmą. Daugiau informacijos rasite skyriuje apie *pakeitimų atšaukimą ir pakartojimą*.

8.2.7 Statistinės suvestinės skydelis

Skydelis *Statistika* (`Ctrl+6`) teikia apibendrintą bet kokio vektorinio sluoksnio informaciją. Šiame skydelyje jūs galite parinkti:

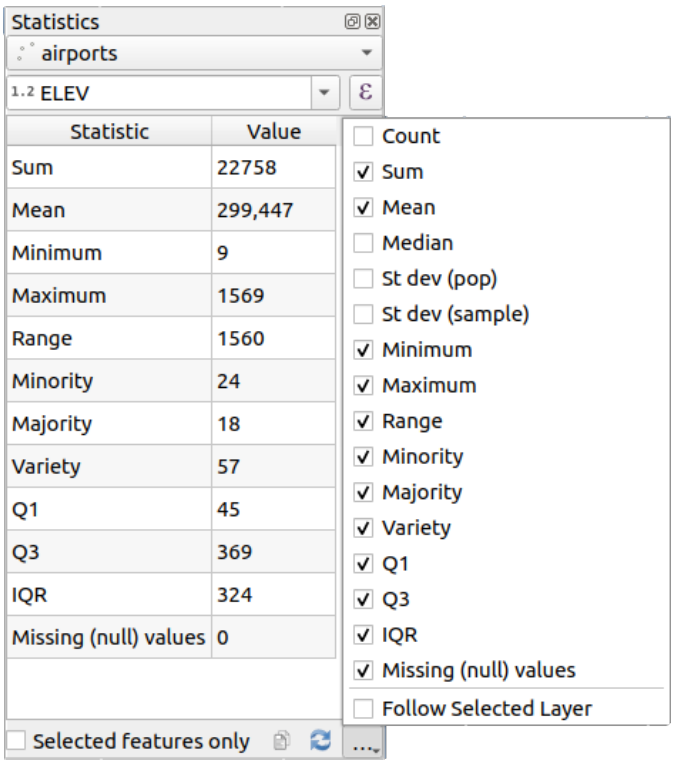
- vektorinį sluoksnį, kuriam reikia skaičiuoti statistiką: jį galima parinkti iš viršutinio iškrentančio meniu arba susieta su aktyviu sluoksniu skydelyje *Sluoksniai*, naudojant varnelę *Sekti pažymėtą sluoksnį*, statistikos apačioje esančiame iškrentančiame meniu
- lauką ar ε *išraišką*, kurią reikia naudoti kiekvienam sluoksniui, paskutinis parinkimas atsimenamas ir automatiškai skaičiuojamas vėl pažymėjus sluoksnį.
- grąžinamą statistiką naudojant iškrentantį mygtuką dialogo apatiniame dešiniame kampe. Priklausomai nuo laukų (ar išraiškos reikšmių) tipo, galimos statistikos yra:

8.2 Table : Kiekvienam lauko tipui galimos statistikos

Statistika	Tekstas	Sveikas skaičius	Realus skaičius	Data
Skaičius	✓	✓	✓	✓
Unikalių reikšmių skaičius	✓			✓
Trūkstamų reikšmių skaičius	✓	✓	✓	✓
Sum		✓	✓	
Vidurkis		✓	✓	✓
Standartinis nuokrypis		✓	✓	
Standartinis nuokrypis pagal mėginį		✓	✓	
Minimali reikšmė	✓	✓	✓	✓
Maksimali reikšmė	✓	✓	✓	✓
Diapazonas		✓	✓	✓
Mažuma	✓	✓	✓	
Dauguma	✓	✓	✓	
Variacija		✓	✓	
Pirmas kvartilis		✓	✓	
Trečias kvartilis		✓	✓	
Tarpkvartilinis diapazonas		✓	✓	
Minimalus ilgis	✓			
Maksimalus ilgis	✓			
Vidutinis ilgis	✓			

Statistikos sumos gali būti:

- grąžinamos visam sluoksniui arba *Tik parinktiems geoobjektams*
- nukopijuotos į iškarpinę ir įkeltos į lentelę kitoje aplikacijoje
- perskaičiuotos naudojant mygtuką kai pasikeičia duomenys (pvz. sukurti nauji ar pašalinti esami geoobjektai, pakeisti atributai)



8.5 Fig. : Rodyti lauko statistiką

8.2.8 Tinkinimo/kūrimo įrankių skydelis

Skydelis *Tinkinimo/kūrimo įrankiai* (`F12`) vienoje vietoje suteikia galimybę valdyti QGIS tinkinimo veiksmus. Teikiami įrankiai organizuojami tokiose kortelėse:

- *API Documentation*: embeds developer oriented documentation
- *Tinklo žurnalas*: įrašo ir rodo tinklo užklausas ir sąveikas
- *Užklausų žurnalas*: įrašo duomenų teikėjų vykdomas SQL komandas ir jų vykdymo laikus
- *Profiliuotojas*: teikia kiekvieno veiksmo įkėlimo laikus, taip randant sulėtėjimo priežastis

Pastaba

Priedų autoriai gali praplėsti šį skydelį savo kortelėmis, skirtomis jų pačių priedų tinkinimui ar kūrimui. Tai daroma naudojant metodą `registerDevToolWidgetFactory`.

API Documentation

The *API Documentation* tab can display either the PyQGIS API documentation or the Qt API documentation. These are embedded in the widget, allowing you to browse and search the docs without leaving QGIS. It is also the place you can display specific help page for objects or classes in your code sample; from the Python interactive console or text editor dialog, select or hover a text and press `F1` or right-click and select Contextual help to get its API help. If you wish to read the active page in the system web browser, then press Open in web browser.

Read *contextual help* for details on the widget configuration.

Tinklo žurnalas

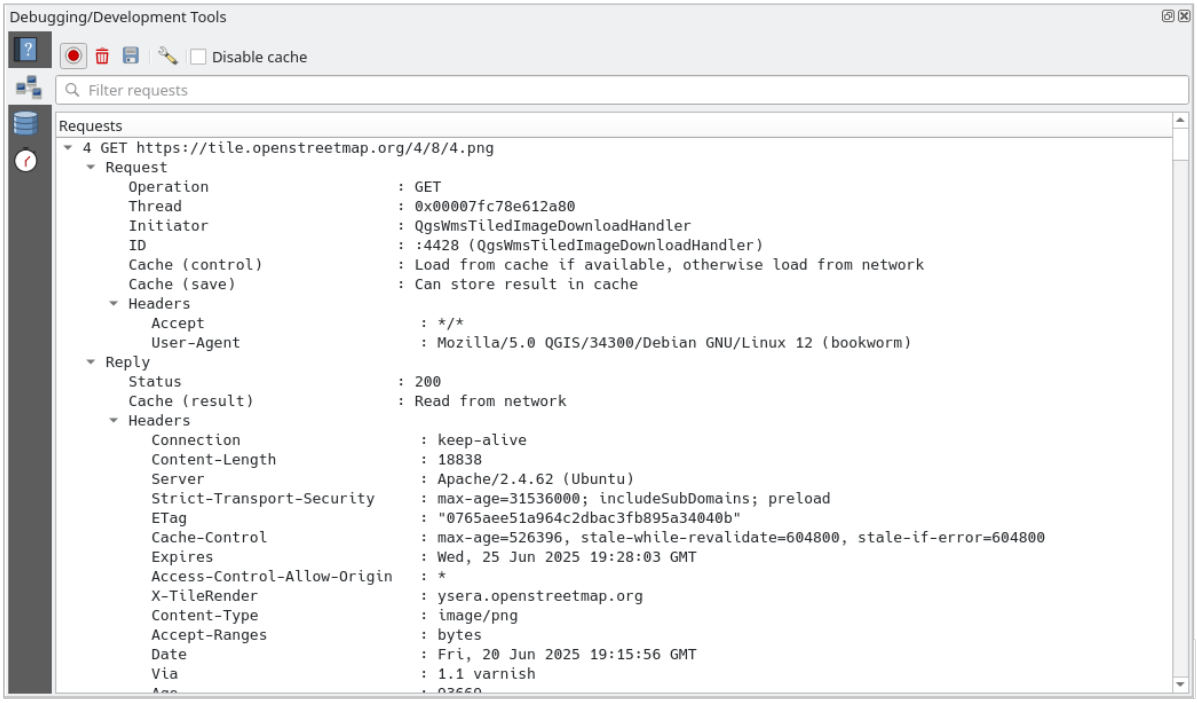
Kortelė *Tinklo žurnalizavimas* leidžia jums įrašyti ir rodyti tinklo užklausas kartu su krūva kitos naudingos informacijos, tokios kaip užklausos ir atsakymo būsena, antraštės, klaidos, SSL konfigūracija, laikai, podėlio būsena ir t.t.

Iš viršutinės įrankinės jūs galite:

- *Įrašyti žurnalą*: kas pradės ar sustabdys žurnalizavimą.
- *Valyti žurnalą*: išvalys žurnalo istoriją.
- *Įrašyti žurnalą…*: iš pradžių parodys didelį įspėjimą, kad žurnalas yra jautrus ir turi būti laikomas konfidencialiu, o tada leis jums įrašyti žurnalą.
- Spauskite iškrentantį meniu *Nustatymai*, kad parinktumėte ar norite *Rodyti sėkmingas užklausas*, *Rodyti laikus* ir *Rodyti atsakymus, paimtus iš podėlio*.
- *Išjungti podėlį*: išjungs podėlį, kad kiekviena užklausa būtų vykdoma.
- *Filtruoti užklausas* pagal URL eilučių iškarpas arba užklausos būseną

Paspaudę dešinį pelės mygtuką ant užklausos jūs galite:

- *Atverti URL*, kuris atvers URL jūsų numatytoje naršyklėje.
- *Kopijuoti URL*
- *Kopijuoti kaip cURL*, kad galėtumėte tai panaudoti terminale.
- *Kopijuoti kaip JSON*: kopijuoja medžio reikšmes kaip json eilutę į iškarpinę, lengvam įkėlimui į rinktų pranešimus ar nutolusiai pagalbai.



8.6 Fig. : GET užklausos tinklo žurnalo išvestis

Užklausų žurnalas

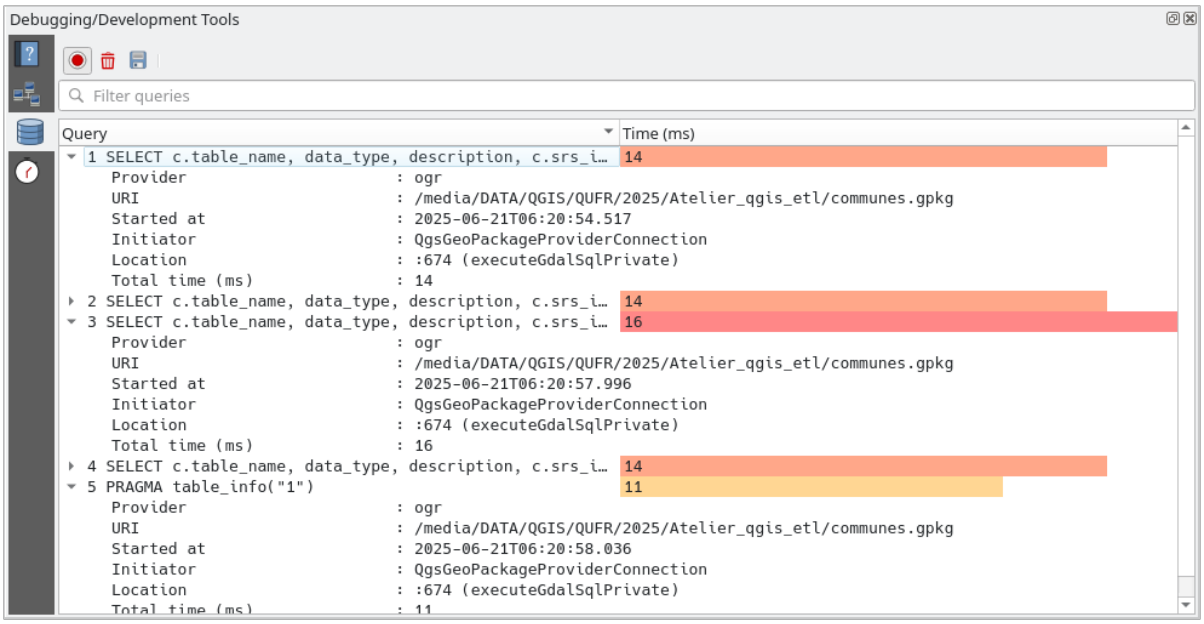
Užklausų žurnalas yra vieta, kur surašomos visos duomenų tiekėjų ir jungčių API į duomenų bazę siunčiamos SQL komandos ir jų vykdymo laikai, pamatuoti QGIS (t.y. kliento, kuris siunčia komandas). Tai gali būti naudinga tiriant greitaveiką ar konkretų sluoksnį QGIS algoritmo ar priedo tinkinimo metu.

Iš viršutinės įrankinės jūs galite:

- *Įrašyti žurnalą*: kas pradės ar sustabdys žurnalizavimą.
- *Valyti žurnalą*: išvalys žurnalo istoriją.
- *Įrašyti žurnalą…*: iš pradžių parodys didelį įspėjimą, kad žurnalas yra jautrus ir turi būti laikomas konfidencialiu, o tada leis jums įrašyti žurnalą.
- *Filtruoti užklausas* pagal užklausos teksto poaibius ar detales, tokias kaip tiekėjo tipas, pradžios laikas, iniciatorius, …

Spauskite praneštą užklausą ir jūs galėsite:

- *Kopijuoti SQL* komandą, išsiųsta iš QGIS į duomenų bazę
- *Kopijuoti kaip JSON*: kopijuoja medžio reikšmes kaip json eilutę į iškarpinę, lengvam įkėlimui į rinktų pranešimus ar nutolusiai pagalbai.



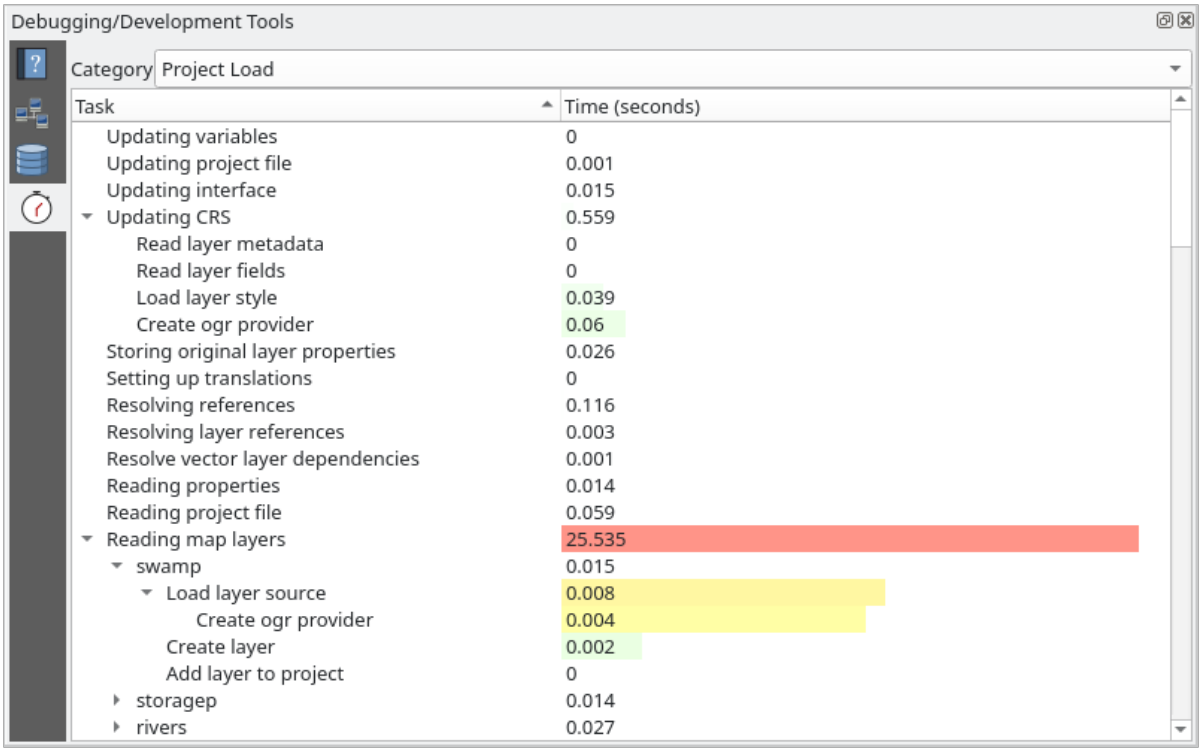
8.7 Fig. : Užklausų žurnalo išvestis

Profiliuotojas

Kortelė *Profiliuotojas* leidžia gauti informaciją, per kiek laiko atliekama kiekviena operacija naudotojo iškviestame veiksme. Priklausomai nuo konteksto, šios operacijos gali būti nustatymų ar meniu skaitymas, žemėlapio drobės ar 3D vaizdo kūrimas, sluoksnių priklausomybių sprendimas, žymelių ar išdėstymų įkėlimas, … Tai padeda identifikuoti lėto įkėlimo laiko priežastis.

Numatytieji palaikomi veiksmai gali būti parinkti iškrentančiame meniu *Kategorija*:

- *Pradžia* padeda diagnozuoti (ir pataisyti) pasitaikančius ilgus QGIS paleidimo laikus.
- *Projekto įkėlimas* leidžia gauti detalią informaciją apie įvairius projekto įkėlimo etapus, kad galėtumėte identifikuoti lėto projekto įkėlimo priežastis.
- *Žemėlapio braižymo įrankis* leidžia identifikuoti skaudančius žemėlapio braižymo taškus ir atsekti, kurie konkrečiai sluoksniai yra lėtų žemėlapio perbraižymų priežastis.



8.8 Fig. : QGIS pradžios profilis

8.3 Išorinių projektų sluoksnių įtraukimas

Kartais prireikia kai kuriuos sluoksnius laikyti skirtinguose projektuose, bet su tuo pačiu stiliumi. Jūs galite arba šiems sluoksniams sukurti *numatytąjį stilių*, arba įtraukti juos iš kito projekto ir taip sutaupyti laiką ir pastangas.

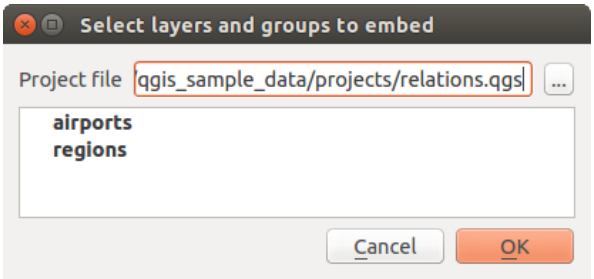
Esamo projekto sluoksnių įtraukimas turi kai kurių pranašumų lyginant su stiliaus naudojimu:

- Gali būti pridėti visų tipų sluoksniai (vektoriniai ir rastro, vietiniai ar internetiniai…)
- Ištraukiant grupes ir sluoksnius jūs galite išlaikyti tą pačią „fono“ sluoksnių medžio struktūrą skirtinguose projektuose
- Nors įtraukti sluoksniai gali būti redaguojami, jūs negalite pakeisti jų savybių, tokių kaip simbologijos, etikečių, formų, numatytųjų reikšmių ir veiksmų, taip užtikrinant nuoseklumą tarp projektų
- Pakeiskite elementus pradiniame projekte ir visi pakeitimai bus išplatinti į visus kitus projektus

Jei norite įtraukti turinį iš kitų projektų failų į jūsų projektą, parinkite *Sluoksnis* ► *Įdėti sluoksnius ir grupes*:

1. Spauskite mygtuką …, kad rastumėte projektą: jūs galite matyti projekto turinį (žr. Fig. 8.9)
2. Laikykite `Ctrl` (ar ✗ `Cmd`) ir spauskite ant sluoksnių ar grupių, kuriuos norite ištraukti
3. Sauskite *Gerai*

Pažymėti sluoksniai ir grupės įtraukiami į skydelį *Sluoksniai* ir rodomi žemėlapio drobėje. Prie jų pavadinimo pridedama piktograma 🔗, kad būtų galima atskirti, o užvedus virš jų pelę bus rodomas kelias iki originalaus projekto.



8.9 Fig. : Parinkite įtraukiamus sluoksnius ir grupes

Kaip bet kurį kitą sluoksnį, įtrauktą sluoksnį galima išimti iš projekto paspaudus ant sluoksnio dešinį pelės mygtuką ir parinkus Išimti.

Patarimas

Įtraukto sluoksnio braižymo pakeitimas

Įtraukto sluoksnio braižymo keisti negalima, nebent jūs pakeitimus darysite pradiniame projekte. Bet jūs galite spausti ant sluoksnio dešinį pelės mygtuką ir parinkti *Kurti sluoksnio dublį* bei taip sukurti sluoksnį, kuris yra pilnavertis ir nebepriklauso nuo pradinio projekto. O tada jau galite saugiai išimti susietą sluoksnį.

8.4 Sąveika su geoobjektais

8.4.1 Geoobjektų pažymėjimas

QGIS teikia kelis įrankius geoobjektų žymėjimui žemėlapio drobėje. Žymėjimo įrankius rasite meniu *Keisti* ► *Pažymėti* arba *Pažymėjimo įrankinėje*.

Pastaba

Pažymėjimo įrankiai veikia su šiuo metu aktyviu sluoksniu.

Rankinis žymėjimas žemėlapio drobėje

Kad pažymėtumėte vieną ar daugiau geoobjektų pele, galite naudoti vieną iš šių įrankių:

- Parinkti geoobjektus pagal plotą ar vienu paspaudimu
- Parinkti geoobjektus poligonu
- Parinkti geoobjektus laisvai
- Parinkti geoobjektus spinduliu

Pastaba

Visi įrankiai, išskyrus Parinkti geoobjektus poligonu, leidžia jums parinkti geoobjektą(us) žemėlapio drobėje vienu paspaudimu.

Pastaba

Naudokite įrankį Parinkti geoobjektus poligonu, jei norite naudoti esamą poligono geoobjektą (iš bet kokio sluoksnio) aktyvaus sluoksnio persidengiančių geoobjektų parinkimui. Spauskite dešiniu pelės mygtuku poligoną ir parinkite jį iš kontekstinio meniu, kuriame rodomas sąrašas visų poligonų, esančių paspaudimo taške. Bus pažymėti visi persidengiantys aktyvaus sluoksnio objektai.

Patarimas

Naudokite meniu *Keisti* ► *Parinkti* ► *Iš naujo parinkti geoobjektus*, kad pakartotumėte jūsų paskutinį parinkimą. Labai naudinga, kai jūs sunkiai dirbote, kad pažymėtumėte daug geoobjektų, o tada netyčia paspaudėte kitur ir išvalėte jūsų pažymėjimą.

Kai naudojate įrankį *Parinkti geoobjektą(us)*, laikykite `Shift` ar `Ctrl` (`Cmd` macOS), kad perjungtumėte, ar geoobjektas parinktas (t.y. arba prideda, arba išima jį iš dabartinio parinkimo).

Kitiems įrankiams galimos skirtingos elgsenos, laikant:

- `Shift`: pridėti geoobjektus į dabartinį parinkimą
- `Ctrl`/`Cmd`: išimti geoobjektus iš dabartinio pažymėjimo
- `Ctrl+Shift`/`Cmd+Shift`: sukirsti su dabartiniu parinkimu, t.y. palikti tik persidengiančius dabartinio parinkimo geoobjektus
- `Alt`: parinkti visus geoobjektus, kurie yra pilnai parinkimo formoje. Naudojant kartu su mygtukais `Shift` ar `Ctrl`/`Cmd` jūs galite pridėti ar išimti geoobjektus į/iš dabartinį parinkimą.

Automatinis parinkimas

Kiti pažymėjimo įrankiai, dauguma iš kurių pasiekiami iš *Atributų lentelės*, vykdo parinkimą pagal geoobjekto atributų arba jo pažymėjimo būseną (pastebėtina, kad atributų lentelė ir žemėlapio drobė rodo tą pačią informaciją, taigi jei jūs parenkate geoobjektą atributų lentelėje, tai jis bus parinktas ir žemėlapio drobėje):

- Select Features By Expression… or press `Ctrl+F3` to select features using the expression dialog
- Parinkti geoobjektus pagal reikšmę… arba spauskite `F3`
- Nužymėti geoobjektus visuose sluoksniuose arba spauskite `Ctrl+Alt+A`, kad nužymėtumėte visus parinktus geoobjektus visuose sluoksniuose
- Nužymėti geoobjektus dabartiniame aktyviame sluoksnyje arba spauskite `Ctrl+Shift+A`
- Parinkti visus geoobjektus arba spauskite `Ctrl+A`, kad parinktumėte visus dabartinio sluoksnio geoobjektus
- Invertuoti geoobjektų parinkimą, kad invertuotumėte dabartinio sluoksnio parinkimą
- Parinkti pagal vietą, kad parinktumėte geoobjektus pagal jų erdvinę sąsają su kitais geoobjektais (tame pačiame arba kitame sluoksnyje - žr. *Select by location*)
- Parinkti pagal atstumą, kad parinktumėte geoobjektus, kurie yra nurodytu atstumu nuo atskaitos geoobjekto - žr. *Select within distance*)

Pavyzdžiui, jei norite rasti regionus, kurių tipas yra „boroughs“ iš QGIS pavyzdinio failo `regions.shp`, jūs galite:

1. Panaudoti piktogramą Parinkti geoobjektus pagal išraišką

2. Išplėsti grupę *Laukai ir reikšmės*
3. Du kartus spausti lauką, pagal kurį norite filtruoti („TYPE_2“)
4. Spausti viršutiniame dešiniame skydelyje *Visos unikalios*
5. Sąraše spausti „Borough“. Redaktoriaus lauke *Išraiška* įrašyti tokią užklausą:

```
"TYPE_2"  =  'Borough'
```

6. Spausti *Parinkti geoobjektus*

Išraiškų kūrimo dialoge jūs taipogi galite naudoti *Funkcijų sąrašas* ► *Paskutinis (Parinkimas)*, kad parinktumėte tai, ką naudojote anksčiau. Dialogas atsimena 20 paskutinių naudotų išraiškų. Daugiau informacijos ir pavyzdžių rasite *Išraiškos*.

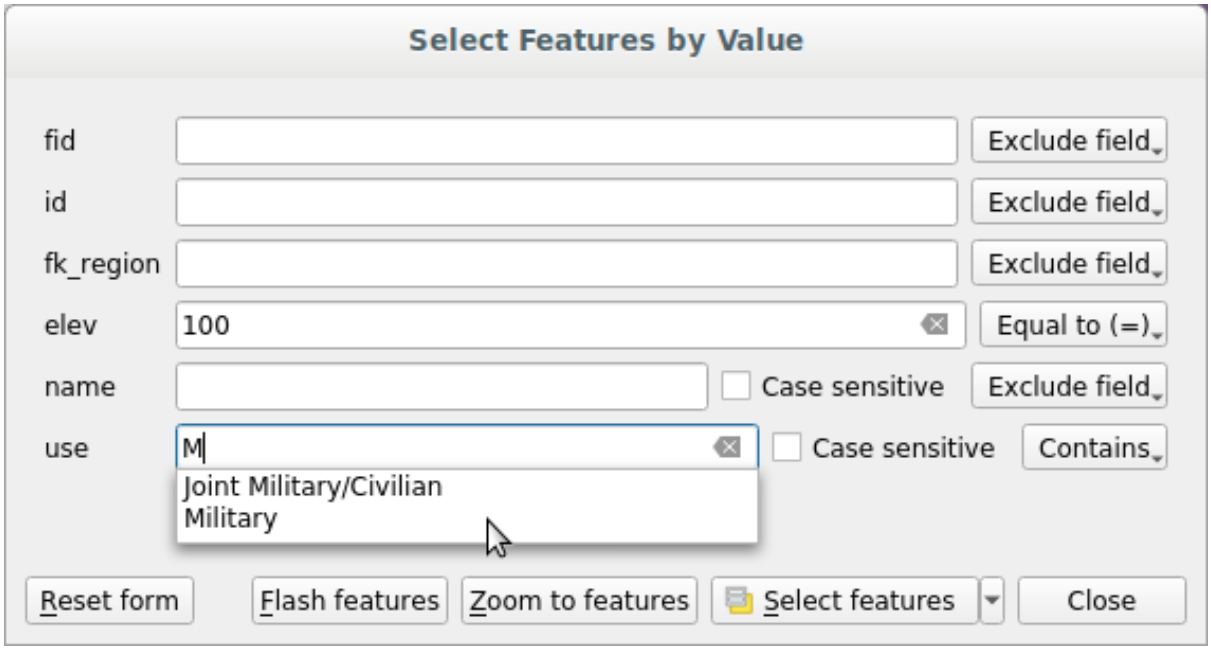
Patarimas

Įrašykite jūsų parinkimą į naują failą

Naudotojai gali įrašyti parinktus geoobjektus į **Naują laikiną juodraštinį sluoksnį** arba **Naują vektorinį sluoksnį** naudojant *Keisti* ► *Kopijuoti geoobjektus* ir tada *Keisti* ► *Įkelti geoobjektus kaip* nurodant norimą formatą.

Parinkti geoobjektus pagal reikšmę

Šis parinkimo įrankis atveria sluoksnio geoobjekto formą ir leidžia naudotojui pasirinkti, kokios reikšmės ieškoti kuriame lauke, ar paiešką atlikti atsižvelgiant į raidžių dydį ir koks turėtų būti naudojamas veiksmas. Įrankis turi automatinį užbaigimą, automatiškai užbaigiantį paieškos laukus esamomis reikšmėmis.



8.10 Fig. : Geoobjektų filtravimas/parinkimas naudojant formos dialogą

Greta kiekvieno lauko yra iškrentantis sąrašas su parinktimis, valdančiomis paieškos elgseną:

8.3 Table : Užklausos veiksmai pagal duomenų tipą

Lauko paieškos variantas	Tekstas	Numeric	Data
Išimti lauką iš paieškos	☑	☑	☑
Lygu (=)	☑	☑	☑
Nelygu (≠)	☑	☑	☑
Daugiau už (>)		☑	☑
Mažiau už (<)		☑	☑
Daugiau arba lygu (≥)		☑	☑
Mažiau arba lygu (≤)		☑	☑
Tarp (imtinai)		☑	☑
Ne tarp (imtinai)		☑	☑
Įtraukia	☑		
Neįtraukia	☑		
Nėra (null)	☑	☑	☑
Yra (not null)	☑	☑	☑
Prasideda	☑		
Baigiasi	☑		

Teksto lyginimams taipogi galima naudoti parinktį ☑ *Atsižvelgti į raidžių dydį.*

Nustačius visas paieškos parinktis, spauskite *Ieškoti geoobjektų*, kad parinktumėte atitinkančius geoobjektus. Iškrentančios parinktys yra:

- *Parinkti geoobjektus*
- *Pridėti į dabartinį parinkimą*
- *Išimti iš dabartinio parinkimo*
- *Filtruoti dabartinį parinkimą*

Taipogi galite išvalyti visas paieškos parinktys naudodami mygtuką *Atstatyti formą.*

Nustačius sąlygas jūs galite:

- *Priartinti geoobjektus* žemėlapio drobėje be poreikio iš naujo parinkti
- *Mirksėti geoobjektus*, paryškinant atitinkančius geoobjektus. Tai patogus būdas identifikuoti objektą jo nerenkant ir nenaudojant identifikavimo įrankio. Pastebėtina, kad mirksėjimas nekeičia žemėlapio drobės apimties, todėl geoobjektai bus matomi tik jei jie yra dabartinėje žemėlapio drobėje.

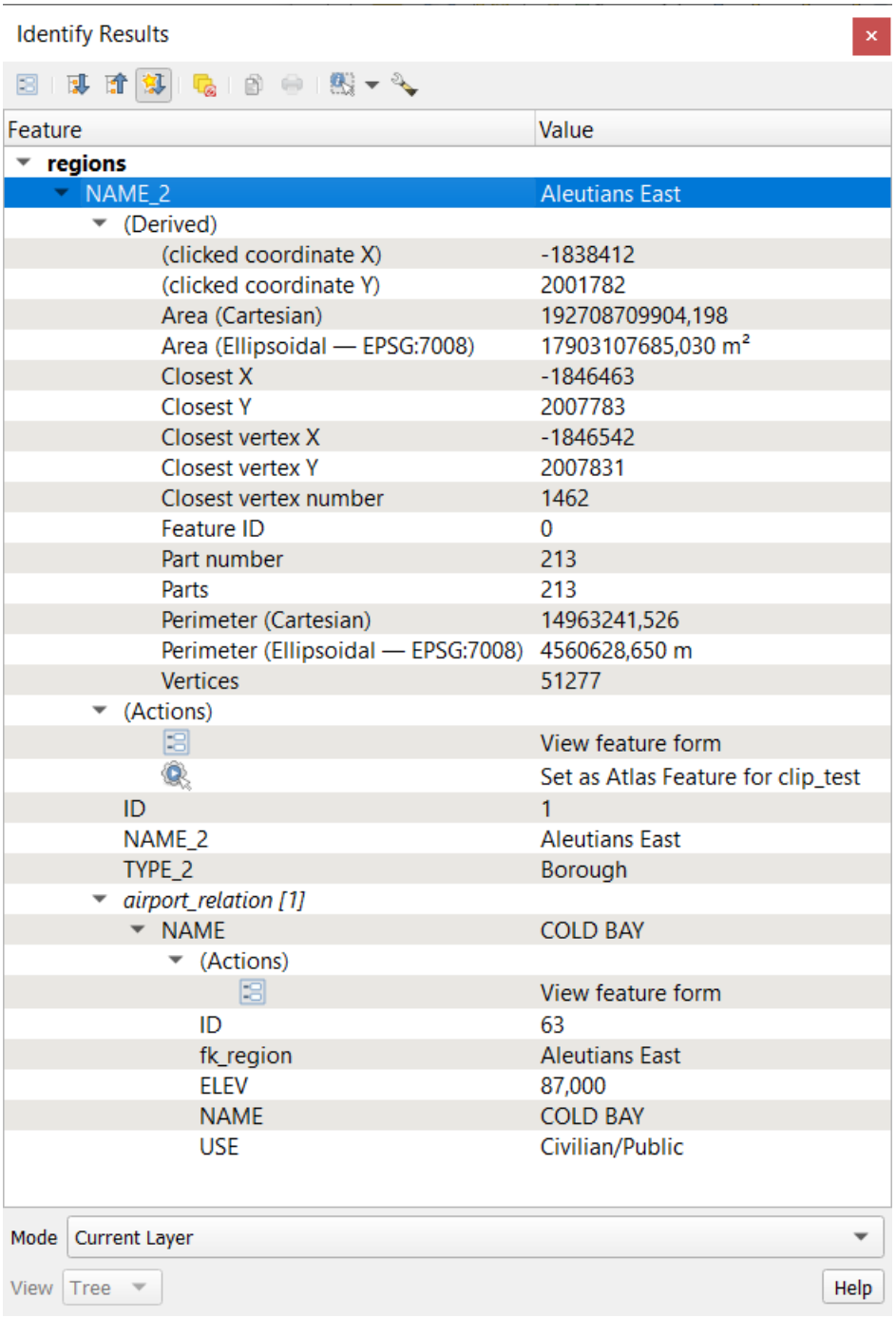
8.4.2 Geoobjektų identifikavimas

Įrankis Identifikuoti geoobjektus leidžia jums sąveikauti su žemėlapio drobe ir gauti informaciją iššokančiame lange apie geoobjektus ar pikselius. Jį galima naudoti gaunant informaciją apie daugumą QGIS palaikomų sluoksnių tipų (vektoriai, rastras, tinklelis, taškų debesis, wms, wfs, …). Norint identifikuoti elementą, naudokite vieną iš:

- *Vaizdas* ► *Identifikuoti geoobjektus*
- `Ctrl+Shift+I` (ar `Cmd+Shift+I`),
- Mygtuką Identifikuoti geoobjektus įrankinėje *Atributai*

Tada spauskite aktyvaus sluoksnio geoobjektą ar pikselį. Identifikuojamas elementas paryškinamas žemėlapio drobėje, kol atveriamas dialogas *Identifikavimo rezultatai* su detalia informacija. Dialogas taipogi rodo kelis mygtukus išmanesnei konfigūracijai.

Identifikavimo rezultatų dialogas



8.11 Fig. : Identifikavimo rezultatų dialogas

Nuo apačios į viršų

- *Vaizdas* valdo bendrą dialogo išvaizdą ir rezultatų formatą, galimi variantai:

  - **Tree**: this is the default view, and returns the results in a tree-structure where the first item is the name of the layer and its children are its identified element(s).
  - **Lentelė**: galimas tik rastro sluoksniams, jis leidžia rodyti rezultatus kaip lentelę, kurios stulpeliai yra `Sluoksnis`, `FID`, `Atributas` ir `Reikšmė`
  - arba **Grafikas**: galimas tik rastro sluoksniams
- *Režimas* leidžia jums pasirinkti sluoksnius, iš kurių bus grąžinami rezultatai. Šie sluoksniai turi būti matomi, jie turi rodyti duomenis žemėlapio drobėje, ir turi turėti nustatytą požymį *identifikuojami* per meniu *Projekto savybės* ► *Duomenų šaltiniai* ► *Sluoksnio galimybės*. Galimi režimai yra:
  - **Dabartinis sluoksnis**: rezultatus grąžins tik sluoksnis(iai), parinktas skydelyje *Sluoksniai*. Jei parinkta grupė, rezultatai bus išrenkami iš visų jos vaikinių sluoksnių.
  - **Nuo viršaus žemyn, sustoti ties pirmu**: rezultatai yra iš aukščiausio geoobjekto ar pikselio po pele.
  - **Iš viršaus žemyn**: rezultatai yra iš sluoksnių su geoobjektu ar pikseliu po pele.
  - **Sluoksnio parinkimas**: atveria kontekstinį meniu, kur naudotojas parenka sluoksnį, kurio geoobjektus reikia identifikuoti. Jei po pele yra tik vienas geoobjektas, rezultatai rodomi automatiškai.
- In the upper part, the information widget: when you identify a data in the map canvas, this is the place where the *Identify Results* dialog will list details about the clicked (or hovered over, depending on the tool in use) items. Their formatting relies on the *selected view*.

  The information displayed by the identify tool will depend on the type of layer you have selected, whether it is:
  - a *vector layer* (including vector tiles or point cloud data),
  - a *raster layer*,
  - a *mesh layer*.
- When in a tree view, the upper part also displays a handful of tools above the results:
  - Atverti formą dabartiniam geoobjektui
  - Išplėsti medį
  - Sutraukti medį
  - Pagal nutylėjimą išskleisti naujus rezultatus, kad nurodytumėte, ar kito geoobjekto identifikavimo informacija turėtų būti suskleista ar išskleista
  - Valyti rezultatus
  - Kopijuoti identifikuotą geoobjektą į iškarpinę, tinka perkėlimui į skaičiuoklę.
  - Spausdinti parinktą HTML atsakymą: tekstinis rezultatų formatavimas spausdinimui ant popieriaus arba įrašymui kaip `.PDF` failu
  - the *interaktyvūs identifikavimo įrankiai*: iškrentantis meniu su įrankiais, skirtais parinkti identifikuojamus geoobjektus ar taškus žemėlapio drobėje
  - Identifikavimo nustatymai leidžia jums aktyvuoti, ar norite:
    - *Automatiškai atverti formą vieno geoobjekto rezultatams*: Įjungus, kiekvieną kartą, kai identifikuojamas vienas geoobjektas, atidaroma forma, kurioje rodomi jo atributai. Tai patogus būdas greitai keisti geoobjekto atributus.
    - *Slėpti rezultatuose išvestinius atributus* jei norite rodyti tik tuos laukus, kurie realiai apibrėžti sluoksnyje
    - *Slėpti rezultatuose NULL reikšmes*
    - *Show Relations* to show relations in the vector layer *Identify results tree*

  - Pagalba leis pasiekti dabartinę dokumentaciją

Geoobjektų identifikavimo įrankio naudojimas

Numatytame vaizde (*Vaizdas: Medis*), skydelis *Identifikavimo rezultatai* teikia kelis įrankius, leidžiančius sąveikauti su užklausiamais sluoksniais. Teisinga šių įrankių kombinacija su *paskirties sluoksnių parinkimu* gali stipriai pagerinti identifikavimo veiksmus:

- Identifikuoti geoobjektą(us) vienu paspaudimu arba paspaudimu ir tempimu
  - vienas paspaudimas arba paspaudimas ir tempimas: grąžinami dengiami geoobjektai iš paskirties sluoksnių
  - dešinys paspaudimas: dengiami geoobjektai iš paskirties sluoksnių surašomi į kontekstinį meniu, sugrupavus pagal sluoksnius. Jūs tada galite pasirinkti:
    * rodyti konkretaus geoobjekto rezultatą,
    * rodyti visų konkretaus sluoksnio geoobjektų rezultatą,
    * vektoriniams sluoksniams taipogi galima atverti jų atributų lentelę atfiltravus pagal grąžintus geoobjektus
    * arba rodyti visus grąžintus geoobjektus.
- Identifikuoti geoobjektus užvedus pelę: judinant pelę žemėlapio drobėje virš paskirties sluoksnio elementų, jie paryškinami ir grąžinami rezultatų skydelyje.
- Identifikuoti geoobjektus poligonu: grąžina elementus, persidengiančius su nubrėžtu arba parinktu poligonu.
  - Nubrėžkite poligoną (spauskite kairį mygtuką, kad padėtumėte tašką, dešinį mygtuką, kad uždarytumėte poligono braižymą) ir visi dengiami geoobjektai iš paskirties sluoksnių bus paryškinti ir grąžinti rezultatų skydelyje.
  - Spauskite dešinį mygtuką ir gausite visų projekte paspaudimo vietoje matomų poligono geoobjektų sąrašą. Parinkite įrašą ir QGIS grąžins visus geoobjektus iš paskirties sluoksnio, kuriuos dengia parinktas poligonas.
- Identifikuoti geoobjektus laisvu apibrėžimu: grąžina elementus, kuriuos dengia laisva ranka nubraižytas poligonas. Nubrėžkite poligoną (kairys paspaudimas pradžiai, tada tempkite žymeklį, kad sukurtumėte plotą, dešiniu mygtuku baikite poligono braižymą). Visi dengiami paskirties sluoksnių geoobjektai paryškinami ir grąžinami rezultatų skydelyje.
- Identifikuoti geoobjektus spinduliu grąžina elementus, kuriuos dengia nubraižytas apskritimas. Nubrėžkite apskritimą (kairiu mygtuku nurodykite centro tašką ir judindami pelės žymeklį suformuokite aplink jį apskritimą arba teksto įvedimo laukelyje įveskite spindulį, tada spauskite dešinį pelės mygtuką arba `Enter`, kad patvirtintumėte apskritimą). Visi padengti paskirties sluoksniu geoobjektai bus paryškinti ir grąžinti rezultatų skydelyje.

Patarimas

Filtruokite sluoksnius Geoobjektų identifikavimo įrankio užklausoms

Per meniu *Projektas* ► *Savybės...* ► *Duomenų šaltiniai* ► *Sluoksnio galimybės*, išjunkite šalia sluoksnio esantį stulpelį *Identifikuojamas*, kad sluoksnis nebūtų užklausiamas naudojant įrankį Identifikuoti geoobjektus. Tai patogus būdas grąžinti geoobjektus tik iš jus dominančių sluoksnių.

Rezultatų kontekstinis meniu

Other functions can be found in the context menu of the identified item. For example, depending on the layer type, you can:

- interact with the individual identified item:
  - *View feature form* or *Edit feature form*, for vector layers
  - *Zoom to feature*: Zooms the map canvas to the extent of the identified feature or pixel
  - *Identify feature*: Restarts and recenters the search results on a *referenced or referencing vector feature* of the identified feature. Only available if *Show Relations* is checked in Identify Settings.
  - *Copy feature*: Copies the vector geometry and attributes, or the pixel geometry
  - *Toggle feature selection*: Adds or removes identified feature to/from the selection
  - *Copy attribute value*: Copies the value in the *Value* column of the tree item you click on
  - *Copy feature attributes*: Copies the attribute names and values of the identified item
  - *Select features by attribute value*: Selects all vector features in the layer that match the selected attribute value
- control the rendering of the layers, the results panel or the map canvas:
  - *Clear result*: Removes results in the window
  - *Clear highlights*: Removes items highlighted on the map
  - *Highlight all*: Highlights in the map canvas all the items identified in the results dialog, regardless of their layer
  - *Highlight layer*: Highlights in the map canvas all the items identified in the layer of the selected entry
  - *Activate layer*: Selects the layer of the identified item in the *Layers* panel
  - *Layer properties*: Opens the layer properties window of the identified item
  - *Expand all* or *Collapse all* the results tree

8.5 Sluoksnio savybių įrašymas ir bendrinimas

8.5.1 Savo stilių tvarkymas

Kai vektorinis sluoksnis pridedamas į žemėlapio drobę, QGIS pagal nutylėjimą braižo jo geoobjektus naudojant atsitiktinį simbolį/spalvą. Bet jūs galite per meniu *Projektas* ► *Savybės...* ► *Numatytieji stiliai* parinkti numatytąjį simbolį, kuris bus taikomas kiekvienam pridėtam sluoksniui priklausomai nuo jo geometrijos tipo.

Bet dažniausiai jūs norėtumėte turėti savo ir sudėtingesnius stilius, kurie būtų (paprasčiau) pritaikomi sluoksniams automatiškai ar rankomis. Jūs galite tai pasiekti naudodami sluoksnio savybių dialogo apačioje esantį meniu *Stilius*. Šis meniu suteikia jums galimybes sukurti, įkelti ir tvarkyti stilius.

Stilius saugo įvairią informaciją, kuri nustatoma sluoksnio ar projekto savybių dialoge ir yra skirta sąveikai su sluoksniu ar jo braižymui. Tai yra:

8.4 Table : Saugomo stiliaus komponentai ir jų aprašai

Kategorija	Vector	Raster
Sluoksnio konfigūracija	*Sluoksnio galimybės*	
	Display Properties	*Display Properties*
Simbologija	*Symbology Properties*	*Symbology Properties* ([1])
3D simbologija	*3D View Properties*	nėra
Užrašai	*Labels Properties*	nėra
Laukai	*Fields Properties* ir *Constraints*	nėra
Atributų forma	*Attributes Form Properties*	nėra
Veiksmai	*Actions Properties*	nėra
Rodymas	*Display Properties*	*Display Properties*
Diagramos	*Diagrams Properties*	nėra
Atributų lentelės konfigūracija	*Darbas su atributų lentele*	
Braižymas	*Rendering Properties*	*Rendering Properties* ([1])
Savo savybės	*Informacinės savybės*	*Informacinės savybės*
Geometrijos parinktys	*Digitizing Properties*	nėra
Ryšiai	*Setting relations between multiple layers*	nėra
Laiko savybės	*Laiko savybės*	*Laiko savybės*
Užrašų nustatymai	*Legend Properties*	nėra
Aukščio savybės	*Aukščio savybės*	*Aukščio savybės*
Pastabos	*Sluoksnio pastabos*	

Pagal nutylėjimą stilius, kuris taikomas įkeltam sluoksniui, pavadintas `default`. Kai pasiekiate idealų ir tinkamą jūsų sluoksnio braižymą, jūs galite įrašyti jį spausdami *Stiliaus* mygtuką ir parinkdami:

- **Pervadinti dabartinį**: Aktyvus stilius pervadinamas ir atnaujinamas dabartinėmis parinktimis
- **Pridėti**: Sukuriamas naujas stilius naudojant dabartines parinktis. Pagal nutylėjimą jis bus įrašomas į QGIS projekto failą. Žemiau rasite informaciją, kaip įrašyti stilių į kitą failą arba duomenų bazę
- **Išimti**: Ištrinti nenorimą stilių, jei turite apibrėžę sluoksniui daugiau nei vieną stilių.

Iškrentančio Stilių sąrašo apačioje matote sluoksniui nustatytus stilius, o aktyvus stilius yra pažymėtas.

Pastebėtina, kad kiekvieną kartą kai jūs tikrinate sluoksnių savybių dialogą, aktyvus stilius atnaujinamas pagal jūsų padarytus pakeitimus.

Galite sukurti sluoksniui kiek tik norite stilių, bet tik vieną galima įjungti vienu metu. Kartu su *Žemėlapio temomis* tai suteikia greitą ir galingą būdą tvarkyti sudėtingus projektus be poreikio kurti sluoksnių kopijas žemėlapių legendoje.

Pastaba

Kadangi jums pritaikant sluoksnio savybių pakeitimus jie laikomi aktyviame stiliuje, visada įsitikinkite, kad keičiate teisingą stilių, kad netyčia nepakeistumėte stilių, naudojamą *žemėlapio temoje*.

[1] For raster layers, symbology and rendering items are melted together, meaning that saving/loading one would also save/load items of/from the other.

Patarimas

Tvarkykite stilius iš sluoksnio kontekstinio meniu

Spauskite dešinį pelės mygtuką ant sluoksnio skydelyje *Sluoksniai*, kad nukopijuotumėte, įkeltumėte ar pervadintumėte sluoksnių stilius. Tiek vektoriniams, tiek rastro sluoksniams jūs galite pasirinkti, kurias stilių kategorijas norite įkelti/kopijuoti. Parinkite *Visos stiliaus kategorijos* arba vieną iš galimų kategorijų, priklausomai nuo sluoksnio tipo. Pastebėtina, kad rastro sluoksnių kategorijos *Simbologija* ir *Braižymas* visada kopijuojamos kartu.

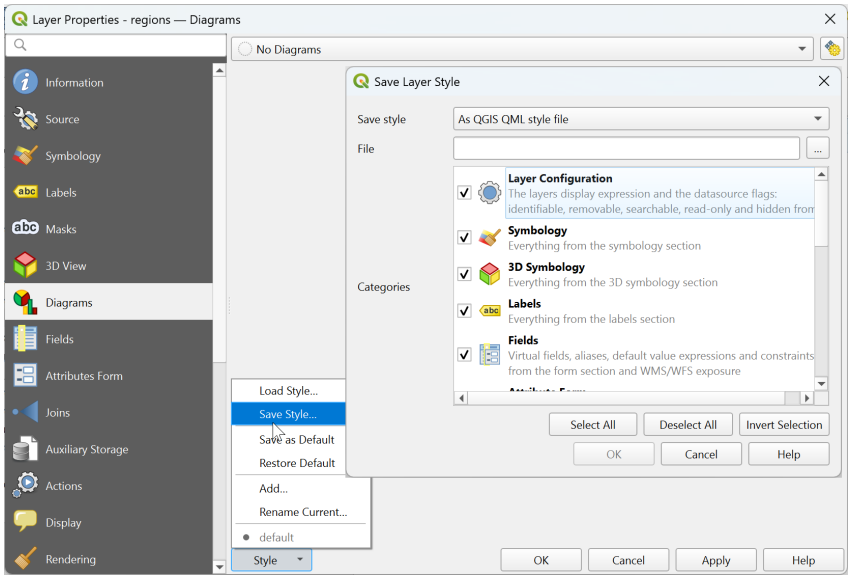
8.5.2 Stilių saugojimas faile ar duomenų bazėje

Nors pagal nutylėjimą stiliai, sukurti per mygtuką *Stilius*, įrašomi į projektą, juos galima nukopijuoti ir įdėti iš vieno sluoksnio į kitą tame pačiame projekte, o taipogi galima įrašyti juos už projekto ribų, kad juos galima būtų įkelti kitame projekte.

Įrašymas į tekstinį failą

Paspaudus *Stilius* ► *Įrašyti stilių* jūs galite įrašyti stilių kaip:

- QGIS sluoksnio stiliaus failą (`.qml`)
- SLD failą (`.sld`), tik vektoriniams sluoksniams



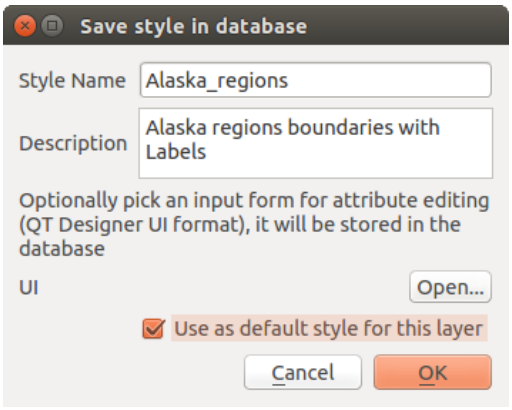
8.12 Fig. : Vektorinio sluoksnio stiliaus parinktys

Kai naudojamas su failais paremtais sluoksniais (`.shp`, `.tab`...), *Įrašyti kaip numatytąjį* sukuria sluoksniui `.qml` failą (su tokiu pačiu pavadinimu). SLD galima eksportuoti iš bet kokio tipo braižymo – vieno simbolio, kategorizuoto, graduoto arba pagal taisykles – bet importuojant SLD sukuriamas arba vieno simbolio arba braižymas pagal taisykles. Tai reiškia, kad kategorizuotas ar graduotas stilius pakeičiamas į paremtą taisyklėmis. Jei norite išlaikyti šiuos braižymus, turite naudoti QML formatą. Kitavertus jis kartais gali būti patogus, kaip lengvas būdas keisti stilių į paremtą taisyklėmis.

Įrašymas į duomenų bazę

Vector layer styles can also be stored in a database if the layer datasource is a database provider. Supported formats are PostgreSQL, GeoPackage, SpatiaLite, MS SQL Server and Oracle. The layer style is saved inside a table (named `layer_styles`) in the database. Click on *Save Style…* ► *Save in database* then fill in the dialog to define a style name, add a description, a `.ui` file if applicable and to check if the style should be the default style.

Galite įrašyti kelis stilius vienoje duomenų bazės lentelėje. Bet kiekviena lentelė turi tik vieną numatytąjį stilių. Numatytuosius stilius galima įrašyti į sluoksnių duomenų bazę arba `qgis.db`, vietinę SQLite duomenų bazę aktyvaus *naudotojo profilio* aplanke.



8.13 Fig. : Stiliaus įrašymo į duomenų bazę dialogas

Patarimas

Dalinimasis stilių failais tarp duomenų bazių

Stilių galite įrašyti į duomenų bazę tik tada, jei sluoksnis ateina iš tokios duomenų bazės. Negalite maišyti duomenų bazių (pavyzdžiui sluoksnis Oracle, o stilius MS SQL Serveryje). Vietoje to naudokite paprasto teksto failą, jei norite, kad stiliumi būtų dalinamasi tarp duomenų bazių.

Pastaba

Jūs galite susidurti su problemomis atstatant `layer_styles` lentelę iš PostgreSQL duomenų bazės atsarginės kopijos. Sekite *QGIS layer_style table and database backup*, kad tai pataisytumėte.

Stiliaus įkėlimas

Įkeliant sluoksnį į QGIS, jei numatytasis šio sluoksnio stilius jau yra, QGIS įkelia sluoksnį su šiuo stiliumi. Taipogi ir *Stilius* ► *Atstatyti numatytąjį* ieško ir įkelia tą failą, pakeičiant dabartinį sluoksnio stilių.

Stilius ► *Įkelti stilių* padeda jums pritaikyti sluoksniui bet kurį įrašytą stilių. Tekstinių failų stilius (`.sld` ar `.qml`) galima taikyti bet kuriam bet kurio formato sluoksniui, o duomenų bazėje įrašytus stilius galima taikyti jei sluoksnis yra iš tos pačios duomenų bazės arba jei sluoksnis laikomas vietinėje QGIS duomenų bazėje.

Dialogas *Duomenų bazės stilių tvarkyklė* rodo sąrašą su sluoksniu susijusių stilių, kurie rasti duomenų bazėje, ir visus kitus joje įrašytus stilius, su jų pavadinimu ir aprašymu.

Patarimas

Greitai dalinkitės sluoksnio stiliumi projekto viduje

Jūs taipogi galite dalintis sluoksnio stiliais projekto viduje neimportuodami failo ar duomenų bazės stiliaus: spauskite dešinį pelės mygtuką ant sluoksnio *Sluoksnių skydelyje* ir iškrentančiame sąraše *Stiliai* nukopijuokite sluoksnio stilių, tada įkelkite jį į grupę ar ar parinktus sluoksnius: stilius pritaikomas visiems to paties tipo (vektorių ar rastro) sluoksniams, kaip ir pradinis sluoksnis. Vektorinių sluoksnių atveju tikrinama, kad sutaptų ir geometrijos tipas (taškas, linija ar poligonas).

8.5.3 Sluoksnio apibrėžimo failas

Sluoksnio apibrėžimus galima įrašyti kaip `Sluoksnio apibrėžimo failus` (`.qlr`) naudojant aktyvaus sluoksnio kontekstinį meniu *Eksportuoti* ► *Įrašyti kaip sluoksnio apibrėžimo failą…*. Sluoksnio apibrėžimo faile (`.qlr`) yra nuorodos į sluoksnių duomenų šaltinius ir jų stiliai. `.qlr` failai rodomi naršyklės skydelyje ir juos galima naudoti, kad pridėtumėte sluoksnius (su jų įrašytu stiliumi) į sluoksnių skydelį. Taipogi galite tempti ir numesti `.qlr` failus iš sistemos failų tvarkyklės į žemėlapio drobę.

8.6 Jūsų duomenų dokumentavimas

Be jūsų duomenų rodymo ir simbolizavimo, QGIS leidžia jums užpildyti:

- **metadata**: information to help people find and understand the dataset, how they can access and use it… these are properties of the datasource and can live out of the QGIS project. Metadata can also be defined for the project file itself.
- **pastabas**: instrukcijas ir komentarus apie sluoksnį dabartiniame projekte

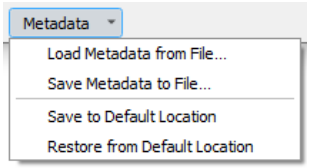
8.6.1 Metaduomenys

In the project or layer properties dialog, the *Metadata* tab provides you with options to create and edit a metadata report on the element. Information to fill concern:

- duomenų *Identifikacija*: bazinės duomenų rinkinio autorių teisės (savininkas, identifikatorius, pavadinimas, aprašymas, kalba…);
- *Kategorijos*, kurioms priklauso duomenys. Kartu su **ISO** kategorijomis, jūs galite nurodyti ir savo;
- *Raktažodžiai* duomenų ištraukimui ir susijusioms sąvokoms pagal standartais paremtą žodyną;
- *Prieiga* prie duomenų rinkinio (licencijos, teisės, mokesčiai ir apribojimai);
- Duomenų aibės *Apimtis*, arba erdvinė (CRS, žemėlapio apimtis, aukščiai), arba laiko;
- Duomenų aibės savininko(ų) *Kontaktai*;
- *Nuorodos* į pagalbinius resursus ir susijusią informaciją;
- duomenų rinkinio *Istorija*.

Užpildytos informacijos santrauka pateikiama kortelėje *Tikrinimas*, kuri padeda jums identifikuoti potencialias su forma susijusias problemas. Tada galite jas pataisyti arba ignoruoti.

Metaduomenys pagal nutylėjimą įrašomi į projekto failą, iškrentantis sąrašas *Metaduomenys* siūlo metaduomenų įrašymo/įkėlimo parinktis iš `.qmd` failo ar „Numatytosios“ vietos.



8.14 Fig. : Metaduomenų įkėlimo/įrašymo parinktys

„Numatytoji“ vieta, kurią naudoja *Įrašyti į numatytąją vietą* ir *Atstatyti iš numatytosios vietos*, keičiasi priklausomai nuo atitinkamo duomenų šaltinio ir jo konfigūracijos:

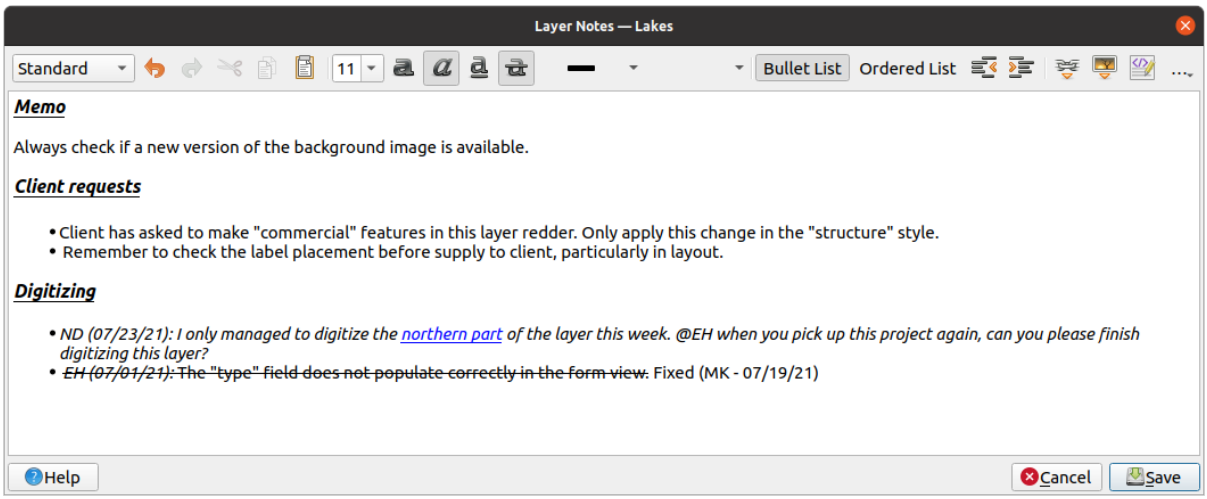
- PostgreSQL duomenų šaltiniams, jei įjungta konfigūracijos parinktis *Leisti įrašyti/įkelti QGIS sluoksnio metaduomenis į duomenų bazę*, metaduomenys saugomi tam skirtoje duomenų bazės lentelėje.
- GeoPackage duomenų šaltiniams *Įrašyti į numatytąją vietą* visada metaduomenis įrašo į vidines GeoPackage metaduomenų lenteles.

  Kai metaduomenys įrašomi į vidines PostgreSQL ar GeoPackage lenteles, pagal juos galima ieškoti ir filtruoti naršyklėje ir *sluoksnio metaduomenų paieškos skydelyje*.
- Visiems kitiems failais paremtiems duomenų šaltiniams *Įrašyti į numatytąją vietą* metaduomenis įrašo į `.qmd` failą greta pagrindinio duomenų failo.
- Visais kitais atvejais *Įrašyti į numatytąją vietą* metaduomenis įrašo į vietinę `.sqlite` duomenų bazę.

8.6.2 Sluoksnio pastabos

Sluoksnio pastabos leidžia dokumentuoti sluoksnį dabartinio projekto viduje. Jose galima laikyti svarbius pranešimus projekto naudotojams, tokius kaip darbų sąrašus, instrukcijas, įspėjimus…

Sluoksnio kontekstiniame meniu skydelyje *Sluoksniai* parinkite *Pridėti sluoksnio pastabas…* ir užpildykite dialoge norimą tekstą.



8.15 Fig. : Sluoksnio pastabų pridėjimas

Dialogas *Pridėti sluoksnio pastabas* teikia html pagrindo kelių eilučių redagavimo įrankį su pilnu rinkiniu įrankių, skirtų:

- teksto valdymui: iškirpimui, kopijavimui, įkėlimui, atšaukimui, pakartojimui
- simbolių formatavimui, taikomam visam ar tik daliai turinio: šrifto dydžiui ir spalva, riebumui, pasvirimui, pabraukimui, perbraukimui, fono spalva, URL paryškinimui

- paragrafo struktūrai: nenumeruotiems ir numeruotiems sąrašams, atitraukimui, iš anksto apibrėžtoms antraštėms
- failo įterpimui, net su pertempimu ir numetimu
- HTML kodo keitimui

Naudodami iškrentantį sąrašą … įrankinės tolimoje dešinėje, jūs galite:

- *Išimti visą formatavimą*
- *Išimti simbolių formatavimą*
- *Išvalyti visą turinį*

Sluoksnių skydelyje sluoksniai su pastaba turi piktogramą , ant kurios užvedus pelę rodoma pastaba. Spauskite piktogramą, jei norite keisti pastabą. Taipogi galite spausti dešinį pelės mygtuką ant sluoksnio ir tada *Keisti sluoksnio pastabą…* ar *Išimti sluoksnio pastabą*.

Pastaba

Pastabos yra *sluoksnio stiliaus* dalis ir gali būti įrašomos į `.qml` ar `.qlr` failą. Jas taipogi galima perkelti iš vieno sluoksnio į kitą kopijuojant sluoksnio stilių.

8.7 Reikšmių saugojimas kintamuosiuose

QGIS jūs galite naudoti kintamuosius naudingų pasikartojančių reikšmių saugojimui (pvz. projekto pavadinimo ar naudotojo pilno vardo). Juos galima naudoti išraiškose. Kintamuosius galima apibrėžti aplikacijos globaliame lygyje, projekto lygyje, sluoksnio lygyje, apdorojimo modeliavimo lygyje, išdėstymo lygyje ir išdėstymo elemento lygyje. Kaip CSS nubanguojančiose taisyklėse, kintamuosius galima permušti - t.y. projekto lygio kintamasis permuš bet kurį aplikacijos lygio kintamąjį su tokiu pačiu pavadinimu. Galite naudoti šiuos kintamuosius kurdami tekstines eilutes ar kitas savas išraiškas naudojant simbolį `@` prieš kintamojo pavadinimą. Pavyzdžiui spausdinimo išdėstyme galima sukurti užrašą su tokiu turiniu:

```
This map was made using QGIS [% @qgis_version %]. The project file for this
map is: [% @project_path %]
```

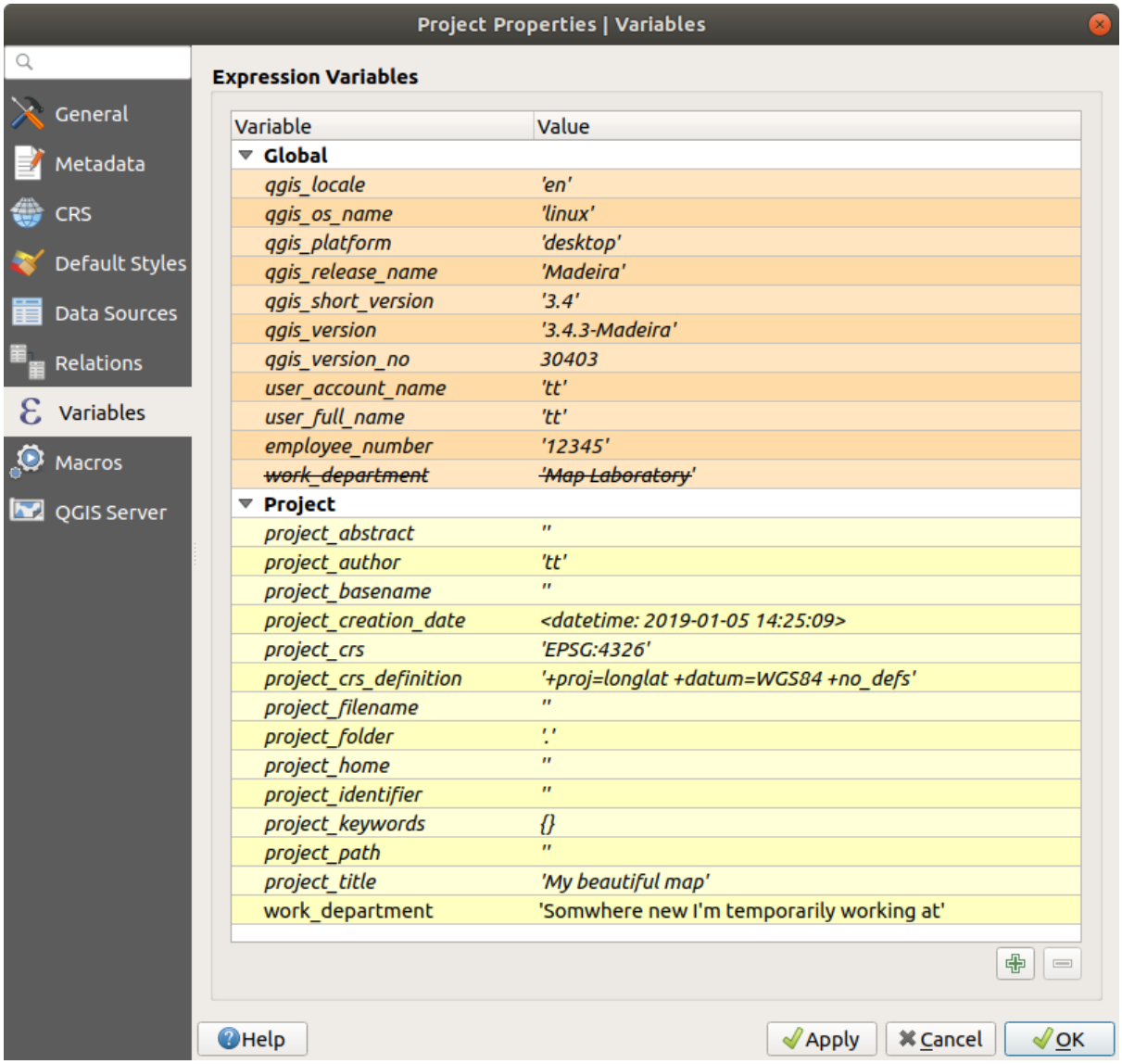
Nubraižys štai tokį užrašą:

```
This map was made using QGIS 3.4.4-Madeira. The project file for this map is:
/gis/qgis-user-conference-2019.qgs
```

Be *iš anksto nustatytų tik skaitymo kintamųjų* jūs galite apibrėžti savo kintamuosius bet kuriam iš aukščiau paminėtų lygių. Jūs galite valdyti:

- **globalius kintamuosius** per meniu *Nustatymai* ► *Parinktys*
- **projekto kintamuosius** dialoge *Projekto savybės* (žr. *Projekto savybės*)
- **vektorinio sluoksnio kintamuosius** dialoge *Sluoksnio savybės* (žr. *Vektorių savybių dialogas*);
- **modeliuotojo kintamuosius** dialoge *Modelių dizaineris* (žr. *Modelių dizaineris*);
- **išdėstymo kintamuosius** skydelyje *Sluoksniai* spausdinimo išdėstyme (žr. *The Layout Panel*);
- ir **išdėstymo elementų kintamuosius** skydelyje *Elementų savybės* spausdinimo išdėstyme (žr. *Layout Items Common Options*).

Siekiant atskirti redaguojamus kintamuosius, tik skaitymo kintamųjų pavadinimai ir reikšmės rodomos pasvirusios. Kitavertus, aukštesnio lygio kintamieji, permušti žemesnio lygio kintamaisiais yra rodomi perbraukti.



8.16 Fig. : Kintamųjų redaktorius projekto lygyje

Pastaba

Daugiau apie kintamuosius jūs galite paskaityti ir rasti pavyzdžių Nyall Dawson tinklaraščio įrašuose Kintamųjų tyrinėjimas QGIS 2.12, 1 dalis, 2 dalis ir 3 dalis.

8.8 Bendri valdikliai

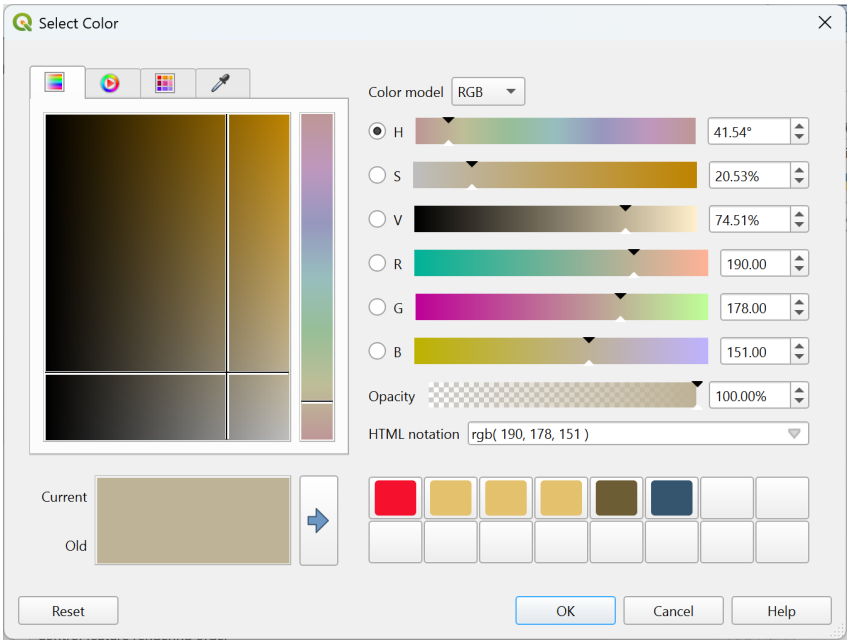
QGIS yra kelios parinktys, su kuriomis pastoviai reikia dirbti. Patogumo dėlei, QGIS teikia jums specialius valdiklius, kurie pristatomi žemiau.

8.8.1 Spalvų parinkiklis

Spalvų dialogas

Dialogas *Parinkite spalvą* bus rodomas visada, kai paspausite piktogramą , kad parinktumėte spalvą. Šio dialogo galimybės priklauso nuo to, kas nurodyta parinktyje *Naudoti savus spalvų parinkimų dialogus*, kurią rasite *Nustatymai* ► *Parinktys...* ► *Bendra*. Jei įjungta, bus naudojamas bendras OS, kuriame veikia QGIS, spalvų dialogas. Priešingu atveju bus naudojamas savas QGIS spalvų dialogas.

Savas spalvos parinkimo dialogas turi keturias skirtingas korteles, kurios leidžia jums parinkti spalvas Spalvų rampoje, Spalvų rate, Spalvų paletėje ar Spalvų parinkikliu. Pirmose dviejose kortelėse jūs galite naršyti po visas galimas spalvų kombinacijas ir pritaikyti jūsų pasirinkimą elementui.

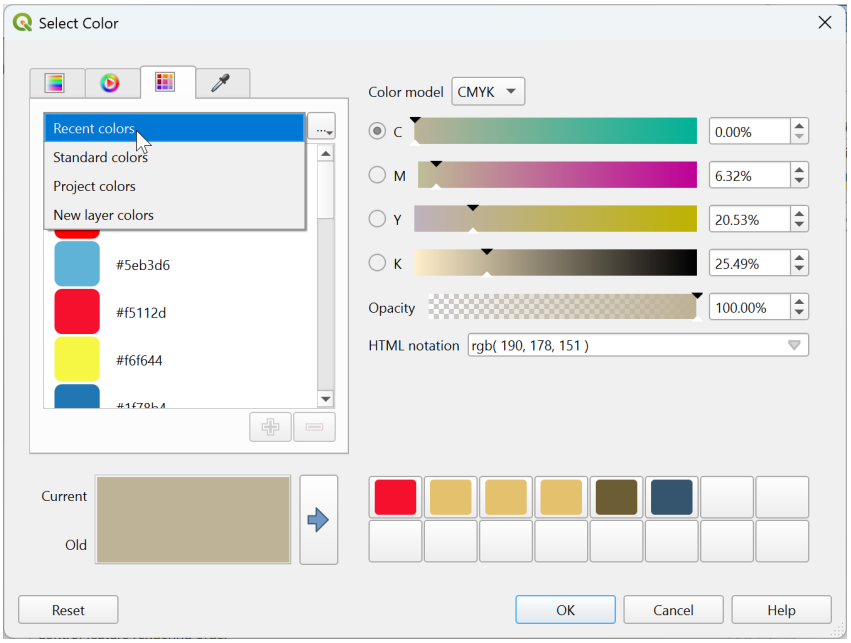


8.17 Fig. : Spalvos parinkimo rampos kortelė

Spalvų pavyzdžių kortelėje jūs galite pasirinkti iš sąrašo spalvų palečių (daugiau informacijos rasite *Spalvų nustatymai*). Visas paletes, išskyrus *Paskutinės spalvos*, galima keisti naudojant mygtukus Pridėti dabartinę spalvą ir Išimti parinktą spalvą, kuriuos rasite po sąrašu.

Šalia iškrentančio paletės sąrašo esantis mygtukas ... taipogi leidžia:

- kopijuoti, įkelti, importuoti ir eksportuoti spalvas
- kurti, importuoti ir išimti spalvų paletes
- pridėti savo paletę į spalvų parinkimo valdiklį su elementu *Rodyti spalvų mygtukuose* (žr. Fig. 8.19)



8.18 Fig. : Spalvų parinkimo pavyzdžių kortelė

Kitas variantas yra naudoti Spalvų parinkiklį, kuris leidžia jums paimti spalvą pelės žymeklio vietoje bet kurioje QGIS UI ar net kitos programos dalyje: spauskite tarpą kol kortelė aktyvi, užveskite pelę virš norimos spalvos ir spauskite ant jos arba dar kartą paspauskite tarpą. Jūs taipogi galite spausti mygtuką *Parinkti spalvą*, kad aktyvuotumėte parinkimą.

Kokį metodą benaudotumėte, parinkta spalva visada bus aprašoma *Spalvų modelio* parinktimis, o modelis pagal nutylėjimą bus toks, koks nurodytas *projekto spalvų modelis*:

- *RGB*: spalvos nustatomos slankikliais, skirtais `HSV` (Spalva, sodrumas ir reikšmė) arba `RGB` (raudona, žalia, mėlyną) reikšmėms.
- *CMYK* (žydra, rožinė, geltona, juoda)

Keisti spalvą yra labai paprasta, tereikia paspausti ant spalvos rato ar rampos ar bet kurio spalvos parametrų slankiklio. Galite keisti tokius parametrus naudodami sukimo valdiklį, esantį greta, arba naudodami pelės ratuką virš atitinkamo slankiklio. Jūs taipogi galite įrašyti spalvą, naudodami HTML notaciją. Na ir galų gale yra slankiklis *Nepermatomumas*, kuris leidžia nurodyti permatomumo lygį.

Dialogas taipogi teikia vizualų palyginimą tarp *Senos* spalvos (pritaikytos objektui) ir *Dabartinės* (t.y. tos, kuri dabar parenkama). Naudodami tempimą ir numetimą ar mygtuką Pridėti spalvą į pavyzdžius, bet kurią iš šių spalvų galima įrašyti į vietą paprastai prieigai.

Patarimas

Greitas spalvos keitimas

Tempkite ir numeskite spalvos parinkimo valdiklį į kitą, kad pritaikytumėte jo spalvą.

Greitas iškrentantis spalvos parinkimas

Spauskite iškrentančio sąrašo rodyklę, esančią dešinėje spalvos mygtuko pusėje, kad parodytumėte valdiklį greitam spalvos parinkimui. Jis suteikia:

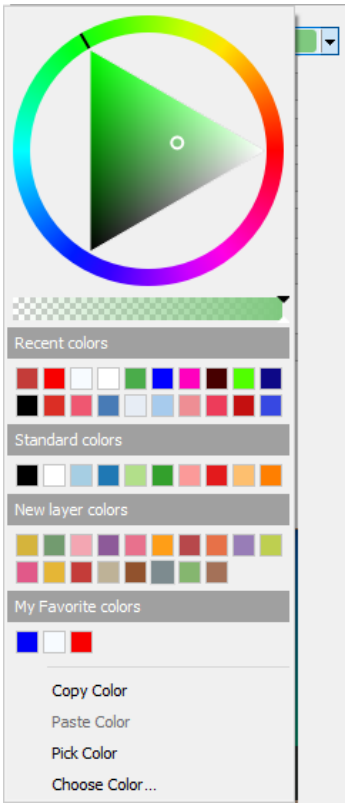
- spalvos ratą, kuriame galite parinkti spalvą
- alfa slankiklį spalvos nepermatomumui keisti
- spalvos paletes, kurioms anksčiau buvo nustatytas *Rodyti spalvų mygtukuose*
- leidžia kopijuoti dabartinę spalvą ir įkelti ją į kitą valdiklį
- parinkti spalvą iš bet kurios jūsų kompiuterio rodomos vietos
- parinkti spalvą spalvų parinkimo dialogu
- galimybę tempti ir numesti spalvą iš vieno valdiklio į kitą greitam pakeitimui

Patarimas

Sukite pelės ratuką virš spalvos parinkimo valdiklio, kad greitai pakeistumėte susijusios spalvos nepermatomumą.

Pastaba

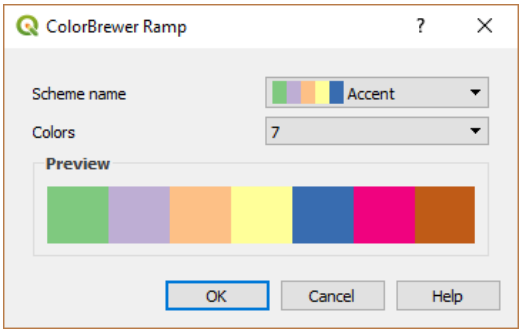
Kai spalvos valdiklis nustatytas į *projekto spalvą* per nuo duomenų priklausančius savybių permušimus, aukščiau išvardintos spalvų keitimo funkcijos negalimos. Jums visų pirma reikės *Atjungti spalvą* ar *Išvalyti* apibrėžimą.



8.19 Fig. : Greito spalvos parinkimo meniu

Iškrentantis spalvų rampos parinkimas

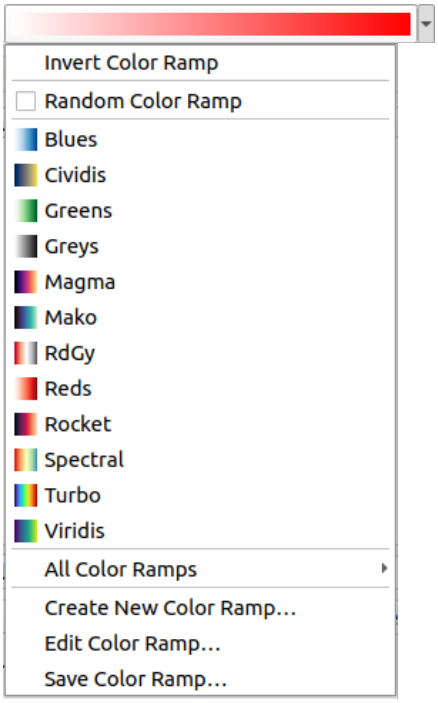
Spalvų rampos - tai praktiškas būdas pritaikyti spalvų aibę vienam ar keliems geoobjektams. Jų kūrimas aprašytas skyriuje *Spalvų rampos kūrimas*. Kaip ir kitoms spalvoms, paspaudus spalvų rampų mygtuką atveriamas atitinkamo spalvos rampos tipo dialogas, kuriame galite keisti jos savybes.



8.20 Fig. : Colorbrewer rampos keitimas

Dešinėje esantis iškrentantis meniu leidžia greitai pasiekti platesnę spalvų rampų aibę ir parinktis:

- *Apsukti spalvų rampą*
- *Išvalyti spalvų rampą* - nuimti bet kurią valdikliui paskirtą spalvų rampą (veikia tik kai kuriuose kontekstuose)
- *Atsitiktinės spalvos*: veikia tik kai kuriuose kontekstuose (pvz., kai spalvų rampa naudojama sluoksnio simbologijai), įjungus šią parinktį, sukuriama ir pritaikoma atsitiktinių spalvų rampa. Tai taipogi įjungia parinktį *Atsitiktinai sudėlioti spalvas*, kad būtų sukurta nauja rampa, jei dabartinė netinka.
- peržiūra `gradient` ir `catalog: cpt-city` spalvų rampų, turinčių žymą **Parankiniai** dialoge *Stilių tvarkyklė*
- *Visos spalvų rampos* leidžia pasiekti visas duomenų bazėje esančias suderinamas spalvų rampas
- *Kurti naują spalvų rampą...* bet kurio palaikomo tipo, kurią galima būtų naudoti dabartiniame valdiklyje (pastebėtina, kad ši spalvų rampa nebus rodoma niekur kitur, nebent jūs įrašysite ją į biblioteką)
- *Keisti spalvų rampą...*, tas pats, kas paspaudus spalvos rampos mygtuką
- *Įrašyti spalvų rampa...*, kad įrašytumėte dabartinę spalvų rampą su jos patinkinimais į stilių biblioteką



8.21 Fig. : Greito spalvos rampo parinkimo valdiklis

8.8.2 Simbolių valdiklis

Simbolių parinkimo valdiklis - tai patogus būdas, kai norite nustatyti geoobjekto simbolio savybes. Paspaudus iškrentančią rodyklę, kartu su *spalvų iškrentančiu valdikliu* rodomas ir šios simbolio parinktys:

- *Konfigūruoti simbolį...*: tas pats, kas paspaudus simbolio parinkimo valdiklį. Atidaromas dialogas, leidžiantis keisti *simbolio parametrus*.
- *Kopijuoti simbolį* iš dabartinio elemento
- *Įdėti simbolį* į dabartinį elementą, pagreitinant konfigūravimą
- *Valyti dabartinį simbolį* leidžia atšaukti simbolį, priskirtą valdikliui (veikia tik kai kuriuose kontekstuose)

Patarimas

Sukite pelės ratuką virš žymeklio ar linijos simbolio valdikliu, kad greitai keistumėte atitinkamo simbolio dydį.

8.8.3 Nutolusio arba įtraukto failo parinkiklis

Greta failų parinkimo valdiklio, mygtukas ... kartais rodys ir rodyklę žemyn. Paprastai taip bus, kai naudojamas:

- SVG failas simboliui ar užrašui
- rastro piešinys simboliams, užrašams, tekstūrai ar dekoracijoms

Paspaudus rodyklę jums bus pateiktas meniu, kuris leis:

- *Parinkti failą...*, kad įkeltumėte failą iš failų sistemos. Failas identifikuojamas keliu, ir QGIS reikia išspręsti kelią, kad parodytų atitinkamą piešinį
- *Iš URL...*, įkelti failą iš nutolusio URL. Kaip ir aukščiau, piešinys bus įkeliamas tik sėkmingai atsiuntus nutolusį resursą

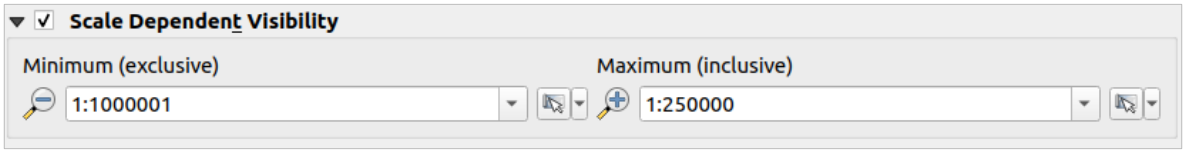
- *Įtraukti failą…*, kad įtrauktumėte failą į dabartinį projektą, stilių duomenų bazę ar spausdinimo išdėstymo šabloną. Tada failas visada braižomas kaip elemento dalis. Tai patogus būdas sukurti savistovų projektą su savo simboliais, kuriais lengva dalintis tarp skirtingų naudotojų ir QGIS diegimų.
- *Ištraukti įdėtą failą…*, kad ištrauktumėte įdėtą failą iš valdiklio ir įrašytumėte jį į diską.

8.8.4 Nuo mastelio priklausomo matomumo parinkimas

Nuo mastelio priklausomo matomumo parinkimas teikia parinktis, valdančias mastelius, kuriuose elementas matomas žemėlapio drobėje. Už nurodyto mastelio diapazono ribų elementai nerodomi. Jis gali būti taikomas pvz. sluoksniams, užrašams ar diagramoms jų savybių kortelėje *Braižymas*.

1. Spauskite varnelę *Nuo mastelio priklausantis matomumas*
2. Į lauką *Minimalus (išskirtinai)* mastelį, kai objektas matomas labiausiai atitolinus, įrašykite rankiniu būdu arba parinkite iš *iš anksto apibrėžtų mastelių*
3. ir/arba lauke *Maksimalus (imtinai)* įrašykite labiausiai priartintą mastelį

   Prie mastelio laukų esantis mygtukas Nustatyti pagal dabartinį drobės mastelį nustato dabartinį drobės mastelį kaip matomumo ribą. Spauskite rodyklę greta mygtuko ir pasiekite mastelius iš išdėstymo žemėlapių ir perpanaudokite juos pildydami laukus.

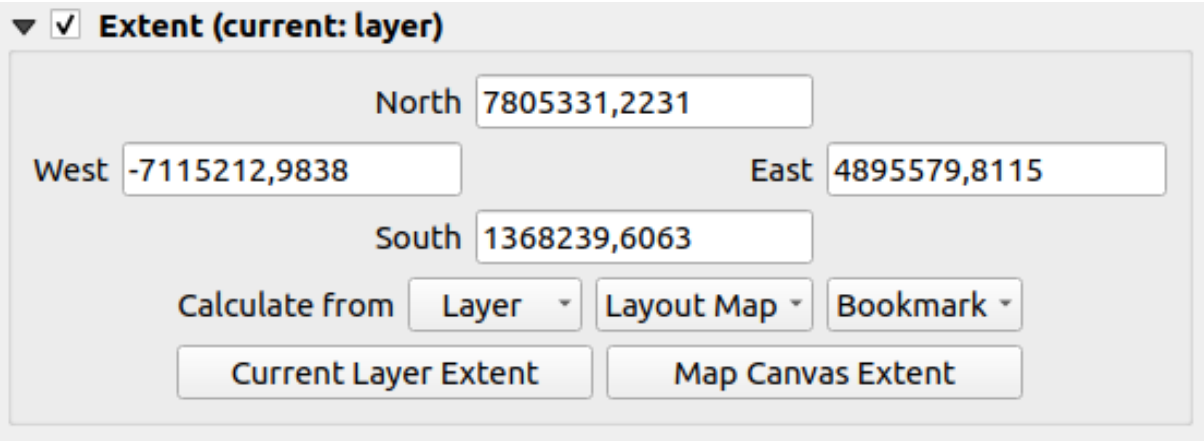


8.22 Fig. : Matomumo mastelių parinkimo valdiklis

8.8.5 Erdvinės apimties parinkimas

Apimties parinkimo valdiklis yra patogus ir greitas būdas parinkti erdvinę apimtį sluoksniui ar apriboti, kur vykdomi veiksmai. Priklausomai nuo konteksto, jis leidžia parinkti:

- *Dabartinio sluoksnio apimtis*: pvz. eksportuojant sluoksnį
- *Skaičiuoti pagal sluoksnį* ►: naudoja dabartiniame projekte įkelto sluoksnio apimtį
- Naudoti dabartinės *Žemėlapio drobės apimtį*
- *Brėžti drobėje*: stačiakampį, kurio kurio koordinatės tada naudojamos
- *Skaičiuoti pagal žymelę*: naudoja įrašytos *žymelės* apimtį
- *Skaičiuoti pagal išdėstymo žemėlapį*: naudoja *išdėstymo žemėlapio* apimtį
- Įvesti arba keisti koordinates kaip `xmin, xmax, ymin, ymax`



8.23 Fig. : Apimties parinkimo valdiklis

8.8.6 Šrifto parinkiklis

Šriftų parinkimo valdiklis yra patogus būdas parinkti tekstinės informacijos šrifto savybes (geoobjektų savybių, dekoracijos užrašų, žemėlapių legendos…). Paspaude iškrentančią rodyklę matysite kelias ar visas iš šių parinkčių:

8.24 Fig. : Šrifto parinkimo iškrentantis meniu

- *Valyti dabartinį teksto formatą* - atšaukti bet kokį valdikliui priskirtą teksto formatą (veikia tik kai kuriuose kontekstuose)
- *Šrifto dydis* susijusiais vienetais
- *Paskutinių šriftų* ► meniu su parinktu aktyviu šriftu (viršuje)
- *Konfigūruoti formatą…*: tas pats, kas paspaudus šrifto parinkimo valdiklį. Atidaro *Teksto formato* dialogą, kurs teikia sudėtingesnes formatavimo parinktis, tokias kaip spalva, permatomumas, orientacija, HTML notacija, buferis, fornas, šešėlis…

- *Kopijuoti formatą* teksto
- *Įkelti formatą* tekstui, pagreitinant konfigūraciją
- the *spalvų valdiklis* greitam spalvos parinkimui

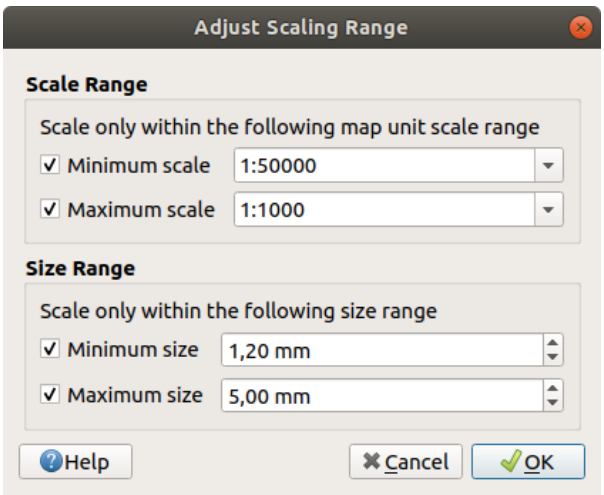
Patarimas

Naudokite pelės ratuką virš šriftų parinkimo valdiklio, kad greitai keistumėte susijusio teksto šrifto dydį.

8.8.7 Vienetų parinkiklis

Elementų (užrašų, simbolių, išdėstymo elementų…) dydžių savybės QGIS nebūtinai susijusios su projekto, ar konkretaus sluoksnio vienetais. Didesnei daliai savybių, iškrentantis *Vienetų* parinkiklis leidžia jums tvarkyti reikšmes pagal jūsų pageidaujamą braižymą (remiantis ekrano raiška, popieriaus dydžiu ar paviršiumi). Galimi vienetai:

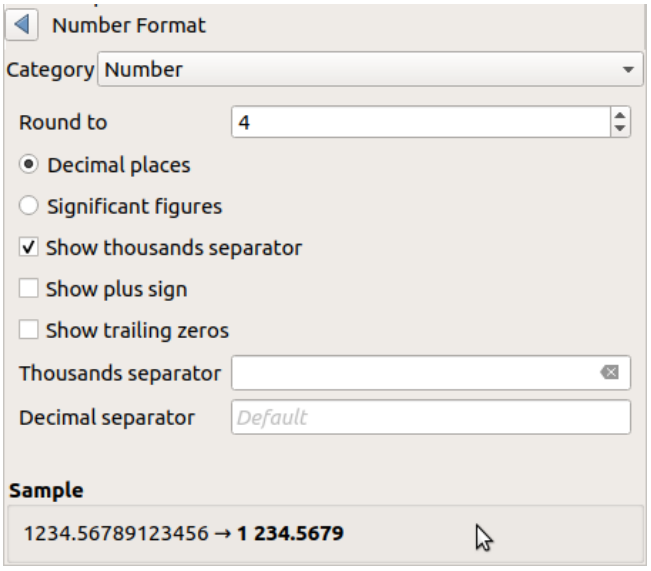
- *Milimetrai*
- *Taškai*
- *Pikseliai*
- *Coliai*
- *Procentai*: leidžia kai kurias savybes nurodyti kaip procentą kitų savybių. Pavyzdžiui tai naudinga kuriant teksto formatus, kur komponentai (buferio dydis, šešėlio spindulys…) gražiai kinta keičiant teksto dydį, vietoje pastovaus buferio/šešėlio dydžio. Tada jums nereikia keisti šių dydžių keičiantis teksto dydžiui.
- *Metrai mastelyje*: Tai leidžia jums visada nustatyti dydį metrais, nepriklausomai nuo to, kokie yra žemėlapio vienetai (pvz. jie gali būti coliai, pėdos, geografiniai laipsniai…). Dydis metrais skaičiuojamas pagal dabartinio projekto elipsoidą ir projekciją kaip metrai dabartinio žemėlapio apimties centre. Projektuotuose žemėlapiuose tai skaičiuojama projektuotais vienetais. Žemėlapiams su geografinėmis koordinatėmis (platuma/ilguma), dydis apytiksliai skaičiuojamas naudojant elipsoidinius skaičiavimus pagal vertikalų žemėlapio mastelį.
- ir *Žemėlapio vienetai*: Dydis skaičiuojamas atsižvelgiant į žemėlapio vaizdo mastelį. Kadangi tai gali reikšti per dideles ar per mažas reikšmes, naudokite greta įvedimo esantį mygtuką 🔧, kad apribotumėte dydžio diapazoną priklausomai nuo:
  - *Minimalaus mastelio* ir *Maksimalaus mastelio*: reikšmė perskaičiuojama pagal žemėlapio vaizdo mastelį kol pasiekiamas vienas iš šių apribojimų. Peržengus ribą naudojama artimiausia ribinė reikšmė.
  - ir/arba *Minimalų dydį* ir *Maksimalų dydį* nurodytą `mm`: reikšmė perskaičiuojama pagal žemėlapio vaizdo mastelį, kol pasiekiamas vienas iš šių apribojimų, tada paliekama apribojimo reikšmė.



8.25 Fig. : Mastelio diapazono perskaičiavimo dialogas

8.8.8 Skaičių formatavimas

Skaičių formatai leidžia formatuoti rodomas skaitines reikšmes, naudojant įvairias formatavimo technikas (pavyzdžiui mokslinė notacija, valiutų reikšmės, procentų reikšmės, savas formatavimas ir pan.). Naudoti galima nustatant tekstą išdėstymo mastelio juostoje ar lentelėje, spalvų rampos legendoje, aukščio profilio brėžinyje, …



8.26 Fig. : Skaitinės reikšmės formatavimas

Palaikomos skirtingos formatų kategorijos. Daugumai iš jų galite nustatyti dalį ar visas iš toliau pateikiamų skaičių parinkčių:

- *Rodyti tūkstančių skyriklį*
- *Rodyti pliuso ženklą*
- *Rodyti nulius pabaigoje*

Bet jie taipogi gali turėti savo nustatymus. Teikiamos kategorijos:

- *Bendra*, numatytoji kategorija: neturi nustatymų ir rodo reikšmes tokias, kokios jos yra tėvinių valdiklių savybėse arba naudojant globalius nustatymus.
- *Savo išraiška*: leidžia susidėlioti savo *QGIS išraišką* skaičių formatavimui. Išraiška gali naudoti kintamąjį `@value`, kad gautų formatuojamą reikšmę ir tada naudoti bet kurią standartinę QGIS išraiškų funkciją, kad suformatuotų skaičių taip, kaip norima.
- *Skaičius*
  - Reikšmę galima *Apvalinti iki* nurodyto skaičiaus *Skaitmenų po kablelio* arba jų *Esminių skaitmenų*
  - tinkinti *Tūkstančių skyriklį* ir *Dešimtainės dalies skirtuką*
- *Kryptis* tekstinei krypties išraiškai naudojant:
  - *Formatą*: galimi reikšmių diapazonai yra nuo `0 iki 180°, su priesaga E/W, -180 iki +180°` ir `0 iki 360°`
  - skaičius *Skaitmenų po kablelio*
- *Valiuta* tekstiniai valiutos reikšmės išraiškai.
  - *Priešdėlis*
  - *Priesaga*
  - skaičius *Skaitmenų po kablelio*

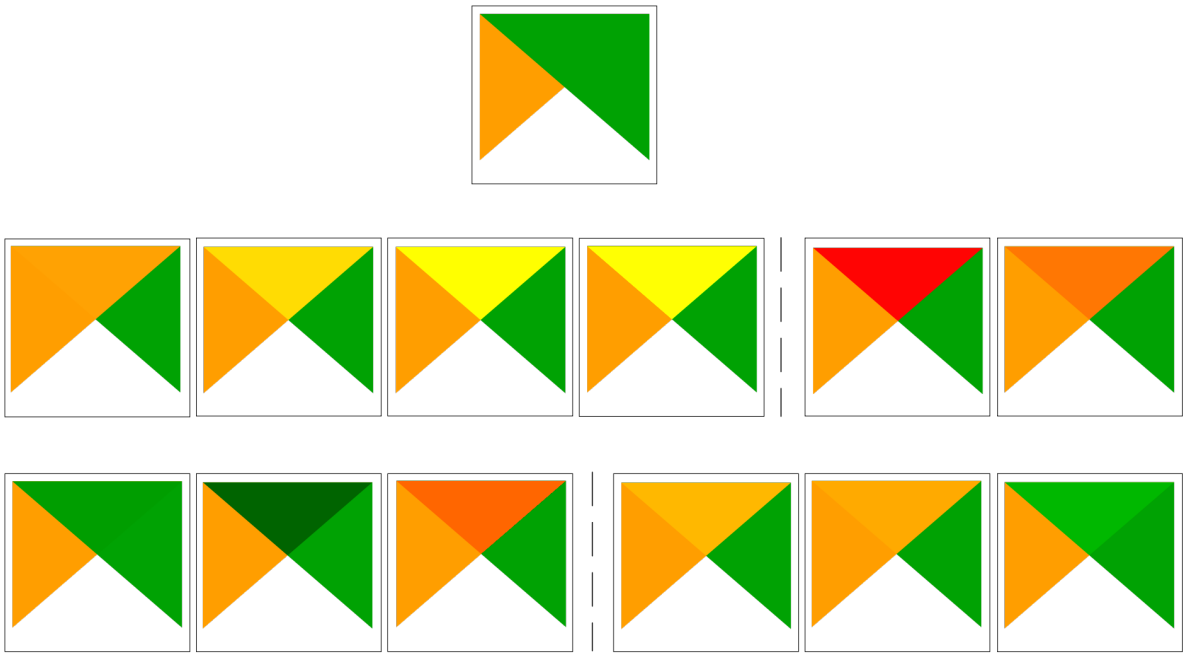
- *Trupmena* paprastai dešimtainės reikšmės trupmeninei išraiškai (pvz. *1/2* vietoje *0.5*)
  - ☐ *Naudoti unikodo viršutinius/apatinius tekstus* rodymui. Pavyzdžiui $^{1}/_{2}$ vietoje 1/2
  - ☐ *Naudoti tam skirtus unikodo simbolius*
  - nurodyti savo *Tūkstančių skirtuką*
- *Procentai* - prideda `%` prie reikšmių, nustatant:
  - skaičius *Skaitmenų po kablelio*
  - *Keitimas* nurodo, ar esamos reikšmės iš karto rodo procentus (tada jos bus paliktos), ar proporciją (tada jos bus pakeistos)
- *Mokslinė* notacija forma `2.56e+03`. Galima nurodyti *Skaitmenų po kablelio* skaičių.

Skiltyje *Pavyzdys* rodoma nustatymų peržiūra.

8.8.9 Suliejimo režimai

QGIS specialiems braižymo efektams siūlo skirtingų suliejimo režimų įrankius, kuriuos jūs anksčiau galėjote matyti tik grafinėse programose. Suliejimo režimus galima taikyti sluoksniams ir geoobjektams bei spausdinimo išdėstymo elementams:

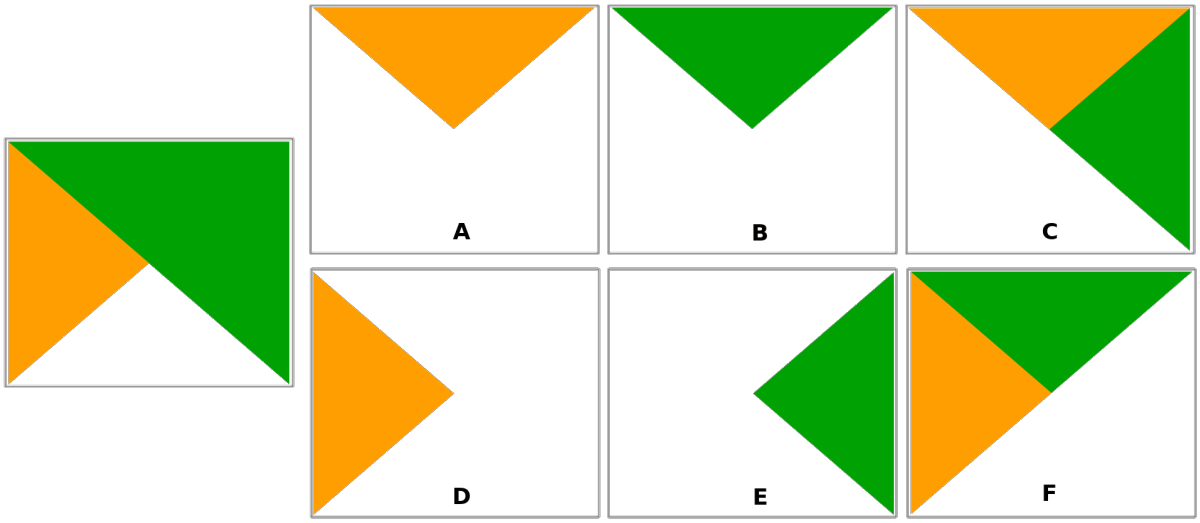
- *Normalus*: tai standartinis suliejimo režimas, kuris naudoja viršutinio pikselio alfa kanalą, kad sulietų jį su žemiau esančiu pikseliu. Spalvos nemaišomos.
- *Šviesinti*: parenka maksimalią kiekvieno komponento reikšmę iš priekyje ir fone esančių pikselių. Turėkite omenyje, kad rezultatai paprastai būna dantyti ir šiurkštūs.
- *Atrinkti*: šviesūs šaltinio pikseliai braižomi virš paskirties, o tamsūs - nebraižomi. Šis režimas naudingiausias maišant vieno elemento tekstūra su kitu elementu (pavyzdžiui naudojant šešėliavimą kito sluoksnio tekstūrai).
- *Vengti*: šviesinti ir sodrinti žemiau esančius pikselius pagal viršuje esančių pikselių šviesumą. Šviesesni viršutiniai pikseliai sukelia apatinių pikselių sodrumo ir ryškumo didėjimą. Tai geriausiai veikia, kai viršutiniai pikseliai nėra per ryškūs. Priešingu atveju efektas bus per ryškus.
- *Sudėtis*: sudeda vieno elemento pikselių reikšmes su kito elemento reikšmėmis. Jei rezultatas viršija maksimalią reikšmę (RGB atveju), rodoma balta. Šis režimas tinka paryškinant geoobjektus.
- *Tamsinti*: išlaiko kiekvieno viršuje ir apačioje esančio pikselio žemiausią reikšmę. Kaip ir šviesinimo atveju, rezultatai būna dantyti ir šiurkštūs.
- *Dauginti*: viršutinio elemento pikseliai dauginami iš atitinkamo apatinio elemento pikselio. Rezultatai būna tamsesni.
- *Deginti*: tamsesnės viršutinio elemento spalvos padaro apatinius elementus tamsesniais. Deginimą galima naudoti tvarkant ir spalvinant apatinius sluoksnius.
- *Perdengti*: kombinuoja dauginimo ir atrinkimo suliejimo režimus. Šviesios dalys tampa šviesesnėmis, o tamsios - tamsesnėmis.
- *Minkšta šviesa*: labai panašus į perdengimą, bet vietoje dauginimo/atrinkimo jis naudoja išdeginimą/vengimą. Tai turėtų imituoti piešinio apšvietimą minkšta šviesa.
- *Stipri šviesa*: stipri šviesa taipogi labai panaši į perdengimo režimą. Ji turėtų imituoti piešinio apšvietimą labai intensyvia šviesa.
- *Skirtumas*: atima viršutinį pikselį iš apatinio pikselio arba atvirkščiai, kad būtų gauta teigiama reikšmė. Suliejimas su juoda reiškia jokio pokyčio, nes visų spalvų skirtumas bus nulis.
- *Atimti*: atima vieno elemento pikselių reikšmes iš kito. Neigiamų reikšmių atveju rodoma juoda.



8.27 Fig. : Suliejimo režimų pavyzdžiai, pritaikyti žaliam geoobjektui virš oranžinio
Nuo viršaus žemyn, iš kairės į dešinę: normalus – šviesinti, atrinkti, vengti, pridėti – skirtumas, atimti – tamsinti, dauginti, išdeginti – perdengimas, švelni šviesa, stipri šviesa

Kai sluoksnis yra grupėje, kuri *braižo sluoksnius kaip grupę*, braižant galimi papildomi suliejimo režimai. Jie leidžia apkirpti vieno sluoksnio nubraižytą turinį kitu - „slėpimo“ - sluoksniu.

- *Paslėpta iš apačios*: išvestis yra viršutinis taškas, kur permatomumas sumažinamas pagal apatinį pikselį.
- *Paslėptas apatinis*: išvestis yra apatinis pikselis, kur permatomumas sumažintas viršutiniu pikseliu.
- *Atvirkščiai paslėpta iš apačios*: išvestis yra viršutinis pikselis, kur permatomumas sumažintas priešingu apatiniu pikseliu.
- *Atvirkščiai paslėptas apatinis*: išvestis yra apatinis pikselis, kur nepermatomumas yra sumažintas atvirkščiu viršutiniu pikseliu.
- *Dažyti viduje apačioje*: viršutinis pikselis yra suliejamas virš apatinio pikselio, o viršutinio pikselio nepermatomumas sumažinamas apatinio pikselio nepermatomumu.
- *Dažyti apačioje viduje*: apatinis pikselis suliejamas su viršuje esančiu pikseliu, o apatinio pikselio nepermatomumas sumažinamas viršutinio pikselio nepermatomumu.



8.28 Fig. : Suliejimo apkarpymo režimų pavyzdžiai, pritaikyti žaliam sluoksniui grupėje
A: Paslėptas apatinis **B**: Paslėpta iš apačios **C**: Dažyti apačioje viduje **D**: Atvirkščiai paslėptas apatinis **E**: Atvirkščiai paslėpta iš apačios **F**: Dažyti viduje apačioje

8.8.10 Nuo duomenų priklausančių permušimų nustatymai

Greta daugelio vektorinių sluoksnių savybių dialogo ar spausdinimo išdėstymo parinkčių jūs rasite piktogramą Nuo duomenų priklausomas permušimas. Naudojant *išraiškas*, kurios remiasi sluoksnio atributais ar elemento nustatymais, standartinėmis ar naudotojo sukurtomis funkcijomis ir *kintamaisiais*, šis įrankis leis jums nustatyti dinamines parametrų reikšmes. Įjungus šią parinktį, šio valdiklio grąžinama reikšmė pritaikoma parametrui nepriklausomai nuo jo normalios reikšmės (varnelė, tekstas, slankiklis…).

Nuo duomenų priklausomo permušimo valdiklis

Paspaudus piktogramą Nuo duomenų priklausantis permušimas, bus rodomi šie įrašai:

- *Aprašymas…* kuris rodo, ar parinktis įjungta, kokios tikimasi įvesties, koks tinkamas įvesties tipas bei dabartinis apibrėžimas. Užvedus pelę ant valdiklio taipogi bus parodyta ši informacija.
- *Saugoti duomenis projekte*: mygtukas leidžia savybę laikyti naudojant mechanizmą *Auxiliary Storage Properties*.
- *Lauko tipas*: įrašas, leidžiantis parinkti vieną iš sluoksnio laukų, kurie atitinka pageidaujamą įvesties tipą.
- *Spalva*: kai valdiklis yra susietas su spalvos savybe, šis meniu duoda prieigą prie spalvų, apibrėžtų kaip dabartinė *projekto spalvų* schema.
- *Kintamasis*: meniu prieigai prie galimų naudotojo apibrėžtų *kintamųjų*
- *Keisti…* mygtukas leidžia kurti arba keisti taikomą išraišką, naudojant dialogą *Išraiškų teksto kūrėjas*. Padedant jums tinkamai užpildyti išraišką, dialoge pateikiamas priminimas apie laukiamą išvesties formatą.
- Mygtukai *Įkelti* ir *Kopijuoti*.
- Mygtukas *Valyti*, leidžiantis išimti nustatymus.
- Skaičių ir spalvų savybėms, *Asistentas…* leidžia nurodyti, kaip skirtinguose masteliuose geoobjekto duomenys taikomi savybėms (daugiau informacijos *žemiau*)

Patarimas

Naudokite dešinį paspaudimą duomenų permušimo (iš)įjungimui

Kai nuo duomenų priklausantis permušimas tinkamai nustatytas, rodoma geltona piktograma ar . Jei jis sulūžęs, piktograma bus raudona ar .

Jūs galite įjungti ir išjungti sukonfigūruotą mygtuką Nuo duomenų priklausantį permušimas tiesiog spausdami valdiklį dešiniu pelės mygtuku.

Patarimas

Naudokite vidurinio mygtuko paspaudimą, kad sukurtumėte arba keistumėte taikomą išraišką

Jūs galite tiesiogiai atverti *Išraiškos teksto kūrėjo* dialogą, kad sukurtumėte ar keistumėte taikomą išraišką, tiesiog paspaudę valdiklį Nuo duomenų priklausantis permušimas viduriniu pelės mygtuku.

Nuo duomenų priklausančio permušimo asistento sąsajos naudojimas

When the Data-defined override button is associated with a size, a width, a rotation, an opacity or a color property, it has an *Assistant…* option that helps you change how the data is applied to the parameter for each feature. Basically, QGIS will scale a range of input values over a range of output values, and render each feature with their corresponding ouput value.

The assistant allows you to:

- Define the *Input* data, i.e.:
  - *Šaltinis*: reprezentuojamas atributas, nurodomas kaip laukas arba *išraiška*
  - the range of values to represent: you can manually enter the values or use the Fetch value range from layer button to fill these fields automatically with the minimum and maximum values returned by the *Source* expression applied to your data. Features whose values are out of the given range will be represented with the property of the closest limit of the output range.
- *Apply transform curve*: by default, output values are applied to input features following a specific scaling method (see below for setting). You can override this logic: enable the transform option, click on the graphic to add break point(s) and drag the point(s) to apply a custom distribution.
- Apibrėžkite *Išvesties* reikšmes: parinktys kinta priklausomai nuo apibrėžiamo parametro. Galite globaliai nurodyti:
  - for color: a linear interpolation of source values is done over a given *color ramp* in order to assign each symbol the correct color. A color to use for symbols with NULL values can also be given.
  - for rotation: a linear interpolation is done over a range of minimum and maximum angles in order to assign each symbol the correct rotation. An angle to use for symbols with NULL values can also be given.
  - for opacity: a polynomial interpolation is done over a range of minimum and maximum opacity values. The *Exponent* field dictates the way source values are mapped to the output range. Power values greater than `1` will cause the output values distribution to start slowly before accelerating as the source values approach their maximum. Smaller exponents (less than `1`) will do the opposite. An opacity value to use for symbols with NULL values can also be given.
  - for width (line symbols): an interpolation is done over a range of minimum and maximum values in order to assign each symbol the correct width. The *Scale method* field controls how the interpolation is done

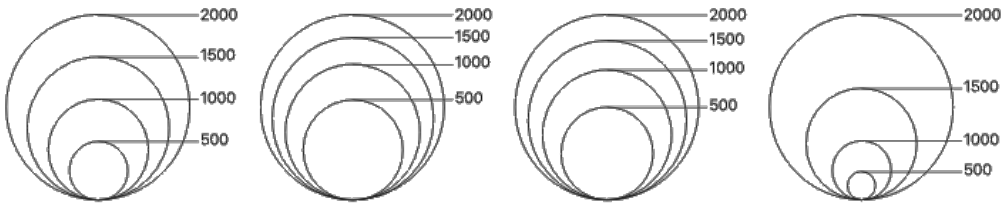
and what the output value represents.

  * **Linear**: linearly interpolates the width of the symbol between the *Size from* and the *Size to* when the *Source* expression value varies from *value from* to *value to*. Symbols width are proportional.
  * **Exponential**: Also interpolates the width, but allows to control the exact power used for scaling through the *Exponent* field. Power values greater than `1` will cause the output values distribution to start slowly before accelerating as the source values approach their maximum. Smaller exponents (less than `1`) will do the opposite.

  The width to use for symbols with NULL values can also be given.

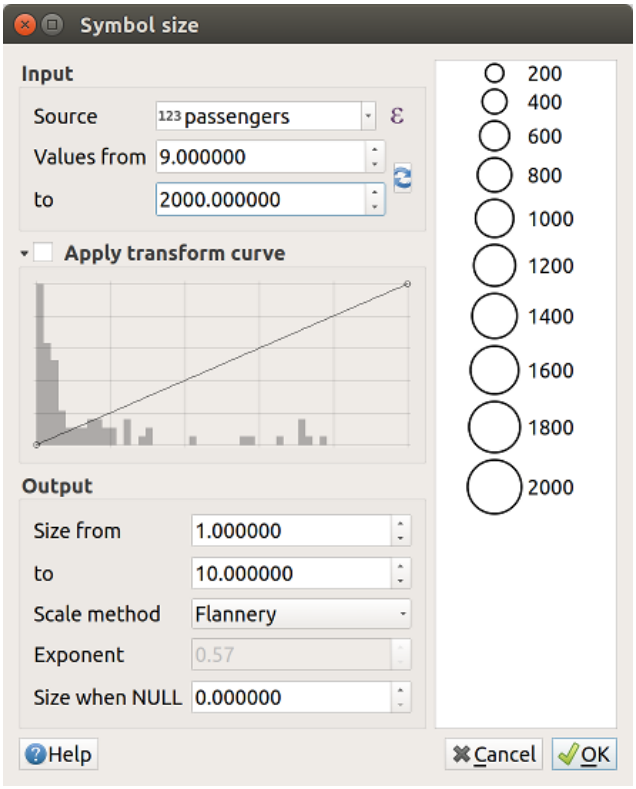
- for size (point symbols): an interpolation is done over a range of minimum and maximum values in order to assign each symbol the correct size „diameter“. The *Scale method* field controls how the interpolation is done and what the output value represents.
  * **Radius**: linearly interpolates the diameter of the symbol between the *Size from* and the *Size to* when the *Source* expression value varies from *value from* to *value to*. Symbols diameter are proportional.
  * **Surface**: scales the symbol diameter according to the square root (power of `0.5`) of the *Source* expression value to linearly interpolate the area of the symbol between the *Size from* and the *Size to* when the *Source* expression value varies from *value from* to *value to*. Symbols area are proportional. This method reflects human perception of size variation better than the **radius** one.
  * **Flannery**: Applies a compensation factor (power of `0.57`) to the area interpolation, in order to mitigate the eyes' perceptual underestimation of proportional areas. Maybe relevant only for circular symbols.
  * **Exponential**: Also interpolates the area, but allows to control the exact power used for scaling through the *Exponent* field. Power values greater than `1` will cause the output values distribution to start slowly before accelerating as the source values approach their maximum. Smaller exponents (less than `1`) will do the opposite.

  The size to use for symbols with NULL values can also be given.



8.29 Fig. : Scaling values using different methods (from left to right): Radius - Surface - Flannery - Exponential of 2

Kai suderinama su savybe, dešinėje dialogo pusėje rodoma gyvai atnaujinama peržiūra, leidžianti jums valdyti reikšmių kitimą.



8.30 Fig. : Geoobjekto dydžio keitimas priklausomai nuo keleivių lauko reikšmės

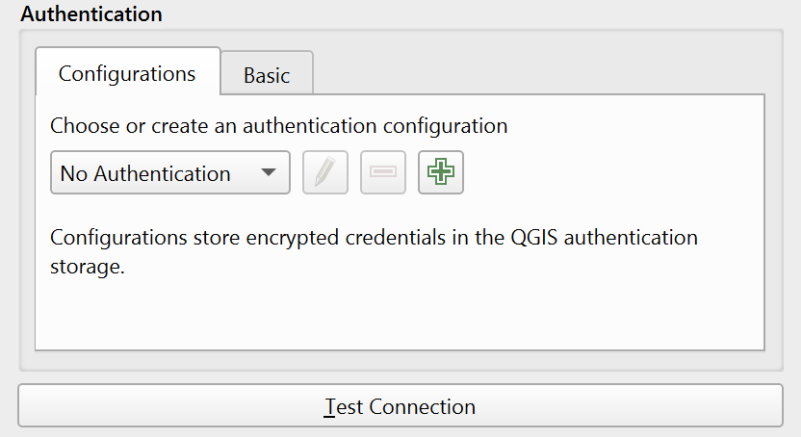
The values presented in the varying size assistant above will set the size ‚Data-defined override' with the editable formula below:

```
coalesce(scale_polynomial("passengers", 9, 2000, 1, 10, 0.57), 0)
```

8.8.11 Autentifikacija

Many data providers in QGIS (such as databases, web services or file-based services) require credentials in order to connect. The authentication widget in QGIS helps users supply these credentials securely, and it offers two ways to do so:

- entering basic credentials directly,
- or selecting an existing authentication configuration stored in the encrypted authentication database.



8.31 Fig. : Authentication widget

Basic

When setting authentication credentials, the *Basic* tab allows you to set:

- *User name*: User name used to connect to the data source.
- *Password*: Password used to connect to the data source.

Convert to configuration will save the user name and password in a new authentication configuration.

You can save any or both of the `User name` and `Password` parameters, in which case they will be used by default each time you need to connect to this database or service. If not saved, you'll be prompted to supply the credentials to connect to the database or service in next QGIS sessions. You can cancel the prompt, in which case the connection will fail. Click the *Ignore for session* button to avoid being prompted again during the current QGIS session. It is also possible to *Ignore for 10 seconds*. This will cancel the prompt but you need to wait 10 seconds before you can attempt to connect again. The connection parameters you entered are stored in a temporary internal cache and returned whenever a username/password for the same database or service is requested, until you end the current QGIS session.

Įspėjimas

QGIS User Settings and Security

In the *Authentication* tab, saving **username** and **password** will keep unprotected credentials in the connection configuration. Those **credentials will be visible** if, for instance, you share the project file with someone. Therefore, it is advisable to save your credentials in an *Authentication configuration* instead (*Configurations* tab - See *Autentifikacijos sistema* for more details) or in a service connection file (see *PostgreSQL Service connection file* for example).

Configurations

QGIS can store credentials under *authentication configurations*. Each configuration safely saves your login details in a portable encrypted database, so you can use them again without typing them every time. These configurations can be safely referenced by their ID tokens in projects and settings.

The authentication configuration selector allows users to choose from existing authentication configurations or create new ones. It provides options to:

- Select an existing authentication configuration from a drop-down list.
- Create a new authentication configuration

- Delete selected configuration
- Edit selected configuration

For more information on the authentication system, see *Autentifikacijos sistema*.

SKYRIUS 9

Pakelkite lygį su išraiškomis

9.1 Išraiškos

Priklausomai nuo sluoksnio duomenų ir standartinių ar naudotojo sukurtų funkcijų, **Išraiškos** siūlo galingą būdą manipuliuoti atributų reikšmėmis, geometrijomis ir kintamaisiais, siekiant dinamiškai geisti geometrijų stilių, užrašų turinį ar vietą, diagramų reikšmes, išdėstymo elementų aukštį, parinkti kai kuriuos geoobjektus, kurti virtualius laukus…

Pastaba

Numatytųjų funkcijų ir kintamųjų, skirtų išraiškų rašymui, sąrašą galima rasti *Funkcijų sąrašas*, kartu su detalia informacija ir pavyzdžiais.

9.1.1 Išraiškų teksto kūrėjas

Pagrindinis išraiškų kūrimo dialogas - *Išraiškų teksto kūrėjas* - pasiekiamas daugelyje QGIS dalių, pavyzdžiui, jį rasite, kai:

- spaudžiate mygtuką ;
- *parenkate geoobjektus* įrankiu Parinkti su išraiška…;
- *keičiate atributus* pavyzdžiui su įrankiu Laukų skaičiuotuvas;
- keičiate simbologiją, užrašus ar išdėstymo elementų parametrus naudodami įrankį Nuo duomenų priklausantis permušimas (žr. *Nuo duomenų priklausančių permušimų nustatymai*);
- kuriate *geometrijos generatoriaus* simbolių sluoksnį;
- vykdote kokį nors *geoapdorojimą*.

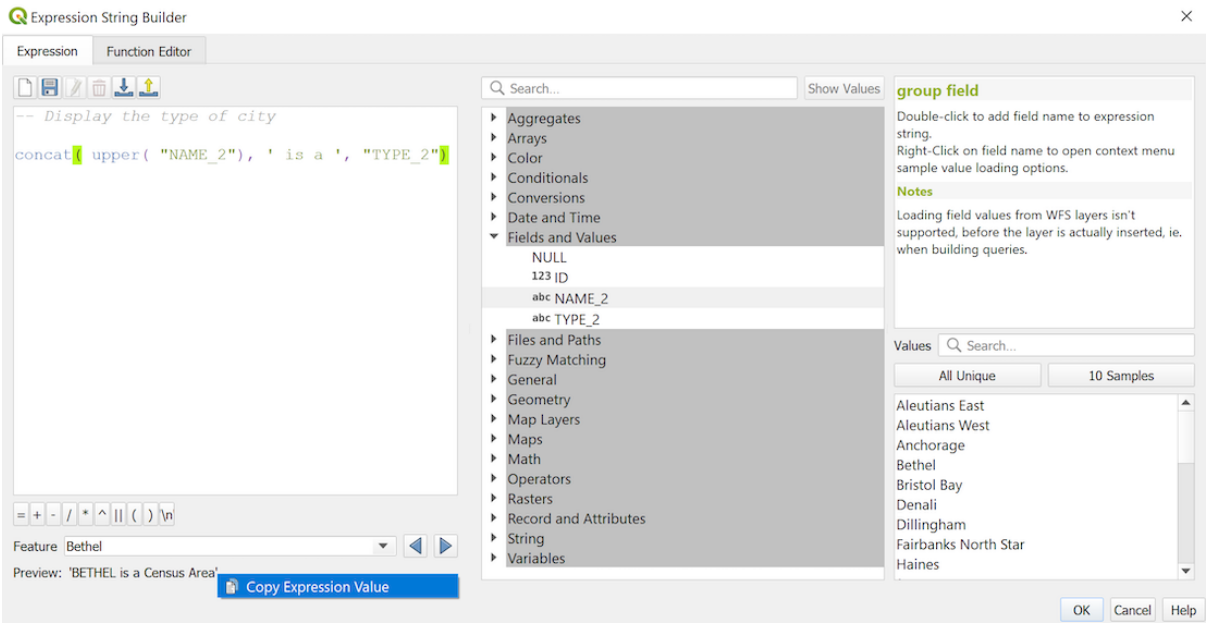
Išraiškų kūrimo dialogas teikia:

- *Išraiškos* kortelę, kuri *standartinių funkcijų* dėka leidžia rašyti ir tikrinti naudojamą išraišką;

- *Funkcijų keitimo kortelę*, kuri padeda išplėsti funkcijų sąrašą kuriant savo funkcijas.

Naudotojo sąsaja

Kortelė *Išraiška* teikia pagrindinę sąsają, skirtą rašyti išraiškas naudojant funkcijas, sluoksnių laukus ir reikšmes. Joje yra šie valdikliai:



9.1 Fig. : Išraiškos kortelė

- Išraiškos rašymo plotas, kuriame rašote ar įkeliate išraiškas. Išraiškų rašymo pagreitinimui galite naudoti automatinį užbaigimą:
  - atitinkamų kintamųjų, funkcijų pavadinimų ir laukų pavadinimų, žemiau esančiame lauke naudokite rodykles `Aukštyn` ir `Žemyn`, kad naršytumėte elementus ir spauskite `Tab`, kad įterptumėte išraišką arba tiesiog spauskite ant norimo elemento.
  - funkcijų parametrų, kurie rodomi juos pildant.

  QGIS taipogi tikrina išraiškų korektiškumą ir paryškina visas klaidas naudojant:
  - *Pabraukimą*: nežinomoms funkcijoms, neteisingiems ar netinkamiems argumentams;
  - *Žymeklį*: kiekvienai kitai klaidai (pvz. trūkstamiems skliausteliams, netikėtiems simboliams) vienoje vietoje.

> **Patarimas**
>
> **Dokumentuokite savo išraišką komentarais**
>
> Kai naudojate sudėtingą išraišką, gera praktika yra pridėti tekstą arba kaip kelių eilučių, arba kaip vidinį komentarą, padedanti jums atsiminti.

```
/*
Labels each region with its highest (in altitude) airport(s)
and altitude, eg 'AMBLER : 264m' for the 'Northwest Artic' region
*/
with_variable(
  'airport_alti', -- stores the highest altitude of the region
  aggregate(
    'airports',
```

```
    'max',
    "ELEV", -- the field containing the altitude
    -- and limit the airports to the region they are within
    filter := within( @geometry, geometry( @parent ) )
  ),
    aggregate( -- finds airports at the same altitude in the region
      'airports',
      'concatenate',
      "NAME",
      filter := within( @geometry, geometry( @parent ) )
        and "ELEV" = @airport_alti
    )
    || ' : ' || @airport_alti || 'm'
    -- using || allows regions without airports to be skipped
)
```

- Virš išraiškų redaktoriaus rasite įrankius, kurie padės jums:
  - Išvalyti išraiškų redaktorių
  - Kurti ir tvarkyti *naudotojo išraiškas*
- Su išraiškų redaktoriumi jūs rasite:
  - Aibę bazinių veiksmų, padedančių kurti išraiškas
  - Tikimosi išvesties formato indikatorių, kai jūs apibrėžiate nuo duomenų priklausančias geoobjektų savybes
  - Gyvą išraiškos *Išvesties peržiūrą* (iki 60 simbolių), pagal nutylėjimą įvertintą su pirmu sluoksnio geoobjektu. Jei norite pažiūrėti išvesties peržiūrą, kuri viršija 60 simbolių, galite užvesti pelę, kad pamatytumėte pilnos išvesties peržiūrą. Norėdami nukopijuoti išvesties peržiūrą į iškarpinę, spauskite ant peržiūros dešinį mygtuką ir parinkite *Kopijuoti išraiškos reikšmę*.

    Galite naršyti ir įvertinti kitus sluoksnio geoobjektus naudodami iškrentantį sąrašą *Geoobjektas* (reikšmės imamos iš sluoksnio savybės *rodomas pavadinimas*).

    Klaidos atveju ji parodoma ir jūs galite gauti daugiau informacijos naudodami pateikiamą nuorodą.
- Funkcijų parinkiklis rodo sąrašą funkcijų, kintamųjų, laukų… sudėliotų į grupes. Yra paieškos laukelis, leidžiantis filtruoti sąrašą ir taip greitai rasti konkrečią funkciją ar lauką. Du kartus paspaudus elementą, jis pridedamas į išraiškų redaktorių.
- Pagalbos skydelis rodo kiekvieno funkcijų parinkiklyje parinkto elemento pagalbą.

> **Patarimas**
>
> Išraiškoje užvedę pelę virš funkcijos pavadinimo spauskite `Ctrl+Paspaudimas`, kad automatiškai parodytumėte dialoge jo pagalbą.

Lauko reikšmių valdiklis, kuris rodomas funkcijų sąraše parinktus lauką, padeda ištraukti geoobjekto atributus:

- ieškant konkrečios reikšmės
- rodant sąrašą *Visų unikalių* ar *10 pavyzdinių* reikšmių. Taipogi pasiekiama paspaudus dešinį pelės mygtuką.

  Kai laukas susietas su kitu sluoksniu ar aibe reikšmių, t.y. jei *lauko valdiklio* tipas yra *RyšioNuoroda*, *ReikšmėsRyšys* ar *ReikšmiųRinkinys*, galima išvardinti visas susijusio lauko reikšmes (iš susijusio sluoksnio, lentelės ar sąrašo). Be to, jūs galite filtruoti šį sąrašą *Rodyti tik naudojamas reikšmes*.

Du kartus paspaudus lauko reikšmę valdiklyje ji bus pridėta į išraiškos redaktorių.

> **Patarimas**
>
> Dešinį skydelį, kuris rodo funkcijų pagalbą ar laukų reikšmes, galima sutraukti (paslėpti) dialoge. Spauskite mygtuką *Rodyti reikšmes* ar *Rodyti pagalbą*, kad jį grąžintumėte.

Išraiškos rašymas

QGIS išraiškos naudojamos parenkant geoobjektus ar nustatant reikšmes. Rašant QGIS išraišką laikomasi kelių taisyklių:

1. **Dialogas nustato kontekstą**: jei jūs įpratę prie SQL, jūs tikriausiai žinote užklausas, tokias kaip *select geoobjektai from sluoksnis where sąlyga* ar *update sluoksnis set laukas = nauja_reikšmė where sąlyga*. QGIS išraiškai irgi reikia visos tokios informacijos, įrankis, kurį jūs naudojate išraiškų redaktoriaus atvėrimui, pateikia dalį jos. Pavyzdžiui, jei turite sluoksnį (`pastatai`) su lauku (`aukštis`) ir tada:

   - spaudžiate įrankį Parinkti pagal išraišką, tai jūs norite atlikti „select geoobjektai from pastatai“. Ir **sąlyga** yra vienintelis dalykas, kurį jums reikia pateikti išraiškos tekstiniame valdiklyje, pvz. įrašykite `"aukštis" > 20`, kad parinktumėte pastatus, kurie yra aukštesni nei 20.
   - atlikus tokį parinkimą, paspaudus mygtuką Laukų skaičiuotuvas ir parinkus „aukštis“ kaip *Keisti esamą lauką*, jūs galite pateikti komandą „update pastatai set aukštis = ??? where aukštis > 20“. Jums belieka tik pateikti **naują reikšmę**, pavyzdžiui tiesiog įveskite `50` išraiškų redaktoriaus tekstiniame laukelyje, kad nustatytumėte prieš tai parinktų pastatų aukštį.

2. **Atkreipkite dėmesį į kabutes**: viengubos kabutės grąžina tekstą, taigi tekstas tarp viengubų kabučių (`'145'`) interpretuojamas kaip eilutė. Dvigubos kabutės grąžina reikšmę, taigi naudokite jas laukams (`"manolaukas"`). Laukus taipogi galima naudoti be kabučių (`manolaukas`). Skaičiams kabučių nereikia (`3.16`).

> **Pastaba**
>
> Funkcijos paprastai kaip argumento laukia teksto ar lauko pavadinimo. Darykite taip:
>
> ```
> attribute( @atlas_feature, 'height' ) -- returns the value stored in the
> ↪"height" attribute of the current atlas feature
> ```
>
> Nedarykite taip:
>
> ```
> attribute( @atlas_feature, "height" ) -- fetches the value of the attribute␣
> ↪named "height" (e.g. 100), and use that value as a field
>                                       -- from which to return the atlas␣
> ↪feature value. Probably wrong as a field named "100" may not exist.
> ```

> **Patarimas**
>
> **Naudokite parametrus su pavadinimais, kad palengvintumėte skaitymą**
>
> Kai kurioms funkcijoms reikia pateikti daug parametrų. Išraiškų variklis leidžia naudoti pavadintus parametrus. Tai reiškia, kad vietoje to, kad rašytumėte mistišką išraišką `clamp( 1, 2, 9)`, galite naudoti `clamp( min:=1, value:=2, max:=9)`. Tai taipogi leidžia sukeisti argumentus vietomis, pvz. `clamp( value:=2, max:=9, min:=1)`. Pavadintų parametrų naudojimas padeda paaiškinti, ką nurodo vienos ar kitos reikšmės, o tai yra naudinga, kai bandote perskaityti išraišką vėliau!

Keli išraiškų naudojimo pavyzdžiai

- Laukų skaičiuotuve paskaičiuokite lauko „pop_density“ reikšmę, naudojant esamus laukus „total_pop“ ir „area_km2“:

```
"total_pop" / "area_km2"
```

- Pažymėti arba kategorizuoti geoobjektus pagal jų plotą:

```
CASE WHEN $area > 10 000 THEN 'Larger' ELSE 'Smaller' END
```

- Pakeisti lauką „density_level“ kategorijomis pagal lauko „pop_density“ reikšmes:

```
CASE WHEN "pop_density" < 50 THEN 'Low population density'
     WHEN "pop_density" >= 50 and "pop_density" < 150 THEN 'Medium population␣
↪density'
     WHEN "pop_density" >= 150 THEN 'High population density'
END
```

- Pritaikyti kategorizuotą stilių visiems geoobjektams pagal tai, ar jų vidutinė būsto kaina yra mažesnė ar didesnė už 10000€ kvadratiniam metrui:

```
"price_m2" > 10000
```

- Naudojant įrankį „Parinkti pagal išraišką…“, parinkite visus geoobjektus, reprezentuojančius „Aukšto gyventojų skaičiaus tankumo“ plotus ir kurių vidutinė būsto kaina yra aukštesnė nei 10000€ už kvadratinį metrą:

```
"density_level" = 'High population density' and "price_m2" > 10000
```

  Ankstesnę išraišką taipogi galima naudoti norint apibrėžti, kuriuos geoobjektus pažymėti ar rodyti žemėlapyje.

- Parinkti geoobjektus, kurie dengia „lands“ sluoksnio natūralią zoną:

```
overlay_intersects( layer:='lands', filter:="zone_type"='Natural' )
```

- Kiekvienam geoobjektui skaičiuoja, kiek juose yra pastatų:

```
array_length( overlay_contains( layer:='buildings', expression:=@id ) )
```

- Kuria sluoksniui skirtingo (tipo) simbolius, naudojant geometrijos generatorių:

```
point_on_surface( @geometry )
```

- Pateiktam taškiniam geoobjektui sukuria uždarą liniją (naudojant `make_line`) aplink jo geometriją:

```
make_line(
  -- using an array of points placed around the original
  array_foreach(
    -- list of angles for placing the projected points (every 90°)
    array:=generate_series( 0, 360, 90 ),
    -- translate the point 20 units in the given direction (angle)
    expression:=project( @geometry, distance:=20, azimuth:=radians( @element )␣
↪)
  )
)
```

- Spausdinimo išdėstymo užraše rodyti „airports“ geoobjektų pavadinimą, kai jie yra išdėstymo elemento „Map 1“ viduje:

```
with_variable( 'extent',
               map_get( item_variables( 'Map 1' ), 'map_extent' ),
               aggregate( 'airports', 'concatenate', "NAME",
                          intersects( @geometry, @extent ), ' ,'
                        )
             )
```

Išraiškų įrašymas

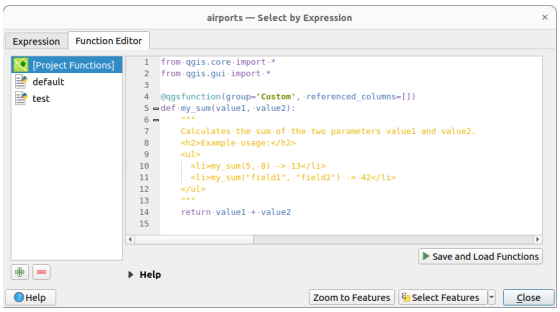
Naudodami virš išraiškų redaktoriaus esantį mygtuką Pridėti dabartinę išraišką į naudotojo išraiškas jūs galite įrašyti svarbias išraiškas, kurias norite greitai pasiekti. Jos bus pasiekiamos grupėje **Naudotojo išraiškos** skydelio viduryje. Jos įrašomos į *naudotojo profilį* (`<userprofile>/QGIS/QGIS3.ini` failas) ir jas galima naudoti visuose išraiškų dialoguose visuose dabartinio naudotojo profilio projektuose.

Virš išraiškų redaktoriaus yra keletas įrankių, kurie padeda tvarkyti naudotojo išraiškas:

- Pridėti dabartinę išraišką į naudotojo išraiškas: įrašyti išraišką į naudotojo profilį. Paprastesniam identifikavimui galima pridėti užrašą ir pagalbos tekstą.
- Keisti parinktą išraišką iš naudotojo išraiškų, taipogi ir jų pagalbą ir užrašą
- Išimti parinktą išraišką iš naudotojo išraiškų
- Importuoti naudotojo išraiškas iš `.json` failo į aktyvaus profilio aplanką
- Eksportuoti naudotojo išraiškas kaip `.json` failą; bendrinamos visos naudotojo profilio faile `QGIS3.ini` esančios naudotojo išraiškos

9.1.2 Funkcijų redaktorius

Kortelėje *Funkcijų redaktorius* jūs galite rašyti savo funkcijas Python kalba. Tai teikia patogų ir paprastą būdą atsižvelgti į poreikius, kurių netenkina standartinės funkcijos.



9.2 Fig. : Funkcijų redaktoriaus kortelė

Norėdami sukurti naują funkciją:

1. Spauskite mygtuką Naujas failas.
2. Parinkite, kur laikyti naują funkciją. Galite pasirinkti įrašyti arba į *Funkcijos failą*, arba į *Projekto funkcijas*.
   - Jei parinksite *Funkcijos failą*, jums reikės įvesti Python `.py` failo pavadinimą, kuris remiasi QGIS šablono failu ir saugomas aktyvaus *naudotojo profilio* aplanko poaplankyje `/python/expressions`, taip jis pasiekiamas visuose projektuose, kuriuos jis atidarys su tuo profiliu.

Savos Python funkcijos, saugomos kaip Funkcijų failai, automatiškai aptinkamos kai pradedate naują projektą su tuo pačiu naudotojo profiliu. Turėkite omenyje, kad bendrinant tokias Python funkcijas jums reikia bendrinti `.py` failą iš aplanko `/python/expressions`.

- Jei parinksite *Projekto funkcijas*, nauja funkcija bus saugoma projekto faile ir bus prieinama tik tame projekte.

  Savos Python funkcijos, kurios saugomos kaip Projekto funkcijos, gali būti lengvai bendrinamos tiesiog bendrinant projekto failą, kuriame jos ir įrašytos. Atveriant projektą su projekto funkcijomis, QGIS gali įkelti arba ignoruoti jas priklausomai nuo to, ar *Įjungtas projekte įtrauktas pythono kodas*.

Naujas elementas pridedamas *Funkcijų redaktoriaus* kortelės kairiame skydelyje.

3. Skydelyje dešinėje rodomas failo turinys: pythono scenarijaus šablonas.
4. Uncomment the code block and start editing it according to your needs. You can rely on code autocompletion, code formatting and typing helpers, as defined in *Pythono nustatymai*. Online API documentation for Qt and PyQGIS objects is also accessible pressing the `F1` key.

   Bendrą informaciją apie @qgsfunction dekoratoriaus nustatymą rasite valdiklyje *Pagalba*.

5. Spauskite mygtuką *Įrašymo ir įkėlimo funkcijos*. Jūsų parašyta funkcija pridedama į funkcijų medį kortelėje *Išraiškos*, pagal nutylėjimą - grupėje `Savo`.
6. Džiaukitės savo nauja funkcija.
7. Jei funkciją reikia tobulinti, įjunkite kortelę *Funkcijų redaktorius*, atlikite pakeitimus ir vėl spauskite mygtuką *Įrašyti ir įkelti funkcijas*, kad jie atsidurtų faile ir taip būtų prieinami bet kurioje išraiškų kortelėje.

Norėdami ištrinti savo funkciją:

1. Įjunkite kortelę *Funkcijų redaktorius*
2. Parinkite funkciją sąraše
3. Spauskite Išimti parinktą funkciją. Funkcija išimama iš sąrašo ir, priklausomai nuo laikymo vietos, atitinkamas `.py` failas ištrinamas iš naudotojo profilio aplanko, arba Python funkcijos išimamos iš projekto failo.

Pastaba

Kai projekto failas su projekto funkcijomis uždaromas, atitinkamos Projekto funkcijos iškeliamos ir jų daugiau nebegalima naudoti QGIS sesijoje.

Pavyzdys

Štai pavyzdys kaip sukurti jūsų nuosavą funkciją `my_sum`, kuri veiks su dviem reikšmėmis.

```
from qgis.core import *
from qgis.gui import *

@qgsfunction(group='Custom', referenced_columns=[])
def my_sum(value1, value2):
    """
    Calculates the sum of the two parameters value1 and value2.
    <h2>Example usage:</h2>
    <ul>
      <li>my_sum(5, 8) -> 13</li>
      <li>my_sum("field1", "field2") -> 42</li>
    </ul>
    """
    return value1 + value2
```

Dekoratorius `@qgsfunction` priima šios argumentus:

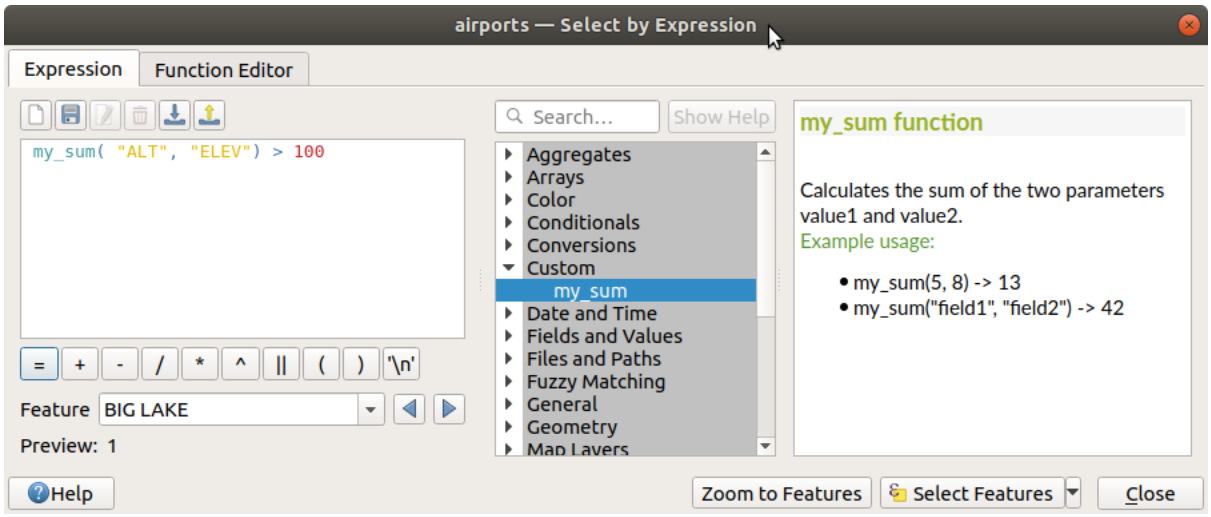
- `group` argumentas nurodo grupę, kurioje funkcija turėtų būti rodoma išraiškų dialoge.
- `usesgeometry`: Nurodykite `True`, jei išraiškai reikia prieigos prie geoobjekto geometrijos. Pagal nutylėjimą `False`.
- `handlesnull`: Nurodykite `True` jei išraiška turi savo NULL reikšmių apdorojimą. Jei `False` (numatyta), rezultatas visada bus NULL, jei bet kuris iš parametrų yra NULL.
- `referenced_columns`: Masyvas atributų pavadinimų, kurie yra privalomi funkcijos vykdymui. Numatyta reikšmė `[QgsFeatureRequest.ALL_ATTRIBUTES]`. Nurodžius dalį laukų ar tuščią sąrašą bus vykdoma greičiau.
- `params_as_list`: Nurodykite `True`, kad perduotumėte funkcijos parametrus kaip sąrašą. Jei `False` (numatytas), funkcija gaus parametrus kaip individualius argumentus.

Pati funkcija priima bet kokį skaičių bet kokio tipo parametrų, kuriuos tik norite perduoti jūsų funkcijai. Jie turėtų būti nurodyti prieš šiuos neprivalomus argumentus:

- `feature`: dabartinis geoobjektas
- `parent`: klasės `QgsExpression` objektas
- `context`: Jei paskutinėje pozicijoje yra argumentas, pavadintas `context`, šiame kintamajame bus klasės `QgsExpressionContext` objektas, kuris duoda prieigą prie įvairios papildomos informacijos, tokios kaip išraiškos kintamieji. Pvz. `context.variable( 'layer_id' )`

Jei šie argumentai nurodyti funkcijos paraše, jie bus automatiškai perduodami funkcijai, be poreikio juos nurodyti išraiškoje.

Ankstesnio pavyzdžio funkciją tada galima naudoti išraiškose:



9.3 Fig. : Savo funkcija, pridėta į išraiškų kortelę

Daugiau savo funkcijų papildomų argumentų pavyzdžių rasite `qgsfunction`. Tolimesnę informaciją apie Pythono kodo kūrimą rasite PyQGIS-Developer-Cookbook.

9.2 Funkcijų sąrašas

QGIS prieinamos funkcijos, operatoriai ir kintamieji išvardinti žemiau, sugrupuoti pagal kategorijas.

9.2.1 Agregavimo funkcijos

Šioje grupėje yra funkcijos, kurios agreguoja sluoksnių ir laukų reikšmes.

aggregate

Grąžina agreguotą reikšmę, paskaičiuotą naudojant kito sluoksnio geoobjektus.

Sintaksė	aggregate(layer, aggregate, expression, [filter], [concatenator=“], [order_by]) [] žymi neprivalomus argumentus
Argumentai	• **layer** - tekstas, reprezentuojantis arba sluoksnio pavadinimą, arba sluoksnio ID • **aggregate** - tekstas, nurodantis, kokį skaičiuoti agregatą. Galimos parinktys yra: – count – count_distinct – count_missing – minimum ar min – maximum ar max – sum – mean – median – stdev – stdevsample – range – minority – majority – q1: pirmas kvartilis – q3: trečias kvartilis – iqr: tarpkvartilinis diapazonas – min_length: minimalus teksto ilgis – max_length: maksimalus teksto ilgis – concatenate: sujungti tekstus jungtuku – concatenate_unique: sujungti unikalius tekstus jungtuku – collect: kurti agreguotą kelių dalių geometriją – array_agg: kurti agreguotų reikšmių masyvą • **expression** - vidinė išraiška arba lauko pavadinimas agregavimui • **filter** - neprivaloma filtro išraiška, ribojanti agregato skaičiavimui naudojamus geoobjektus. Laukai ir geometrijos yra iš susieto sluoksnio geoobjektų. Šaltinio geoobjektą galima pasiekti kintamuoju @parent. • **concatenator** - neprivalomas tekstas, kuris naudojamas jungiant reikšmes, kai vykdomi „concatenate“ ir „concatenate_unique“ agregatai • **order_by** - neprivaloma filtro išraiška agregato skaičiavimo geoobjektų rikiavimui. Laukai ir geometrija yra iš susiejamo sluoksnio geoobjektų. Pagal nutylėjimą geoobjektai grąžinami atsitiktine tvarka.
Pavyzdžiai	• `aggregate(layer:='stotys',aggregate:='sum', expression:="keleiviai")` → visų sluoksnio stotys lauko keleiviai reikšmių suma • `aggregate('stotys','sum', "keleiviai"/7)` → skaičiuoja „keleiviai“ dienos vidurkį, dalinant lauko „keleiviai“ reikšmę iš 7, prieš susumuojant reikšmes • `aggregate(layer:='stotys',aggregate:='sum', expression:="keleiviai",filter:="klase">3)` → sumuoja lauko „keleiviai“ reikšmes tik iš geoobjektų, kurių atributas „klase“ yra didesnis už 3 • `aggregate(layer:='stotys',aggregate:='concatenate', expression:="pavadinimas", concatenator:=',')` → sąrašas visų sluoksnio stotys geoobjektų pavadinimų, atskirtų kableliais • `aggregate(layer:='salys', aggregate:='max', expression:="kodas", filter:=intersects( @geometry, geometry(@parent) ) )` → Sluoksnio „salys“ laukas „kodas“ iš susikertančios šalies • `aggregate(layer:='stotys',aggregate:='sum', expression:="keleiviai",filter:=contains( @atlas_geometry, @geometry ) )` → suma visų sluoksnio „stotys“ lauko „keleiviai“ reikšmių, esančių dabartiniame atlaso geoobjekte • `aggregate(layer:='stotys', aggregate:='collect', expression:=centroid(@geometry), filter:="regionas" = attribute(@parent,'name') )` → agreguoja visas stočių geometrijas, esančias tame pačiame regione, kaip ir dabartinis geoobjektas

array_agg

Grąžina lauko ar išraiškos reikšmių agreguotą masyvą.

Sintaksė	array_agg(expression, [group_by], [filter], [order_by]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui • **order_by** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų rikiavimui. Pagal nutylėjimą geoobjektai grąžinami atsitiktine tvarka.
Pavyzdžiai	• `array_agg("pavadinimas",group_by:="busena")` → sąrašas pavadinimo reikšmių, sugrupuotas pagal būsenos lauką

collect

Grąžina pagal išraišką gautų geometrijų agreguotą kelių dalių geometriją

Sintaksė	collect(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojama geometrijos išraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `collect( @geometry )` → agreguotų geometrijų kelių dalių geometrija • `collect( centroid(@geometry), group_by:="region", filter:= "use" = 'civilian' )` → pagal regiono reikšmę ištrauktų civilinių geoobjektų agreguoti centroidai

concatenate

Grąžina visas agreguotas eilutes iš lauko ar išraiškos, sujungtos skirtuku.

Sintaksė	concatenate(expression, [group_by], [filter], [concatenator], [order_by]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui • **concatenator** - neprivalomas tekstas, kurį reikia naudoti jungiant reikšmes. Pagal nutylėjimą tuščia. • **order_by** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų rikiavimui. Pagal nutylėjimą geoobjektai grąžinami atsitiktine tvarka.
Pavyzdžiai	• `concatenate("miesto_pavadinimas",group_by:="valstybė", concatenator:=',')` → kableliais atskirtas sąrašas miestų pavadinimų, sugrupuotų pagal valstybės lauką

concatenate_unique

Grąžina unikalias eilutes iš laukų ar išraiškų, sujungtų skirtuku

Sintaksė	concatenate_unique(expression, [group_by], [filter], [concatenator], [order_by]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui • **concatenator** - neprivalomas tekstas, kurį reikia naudoti jungiant reikšmes. Pagal nutylėjimą tuščia. • **order_by** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų rikiavimui. Pagal nutylėjimą geoobjektai grąžinami atsitiktine tvarka.
Pavyzdžiai	• `concatenate_unique("miesto_pavadinimas",group_by:="šalis", concatenator:=',')` → kableliais atskirtas sąrašas unikalių miesto pavadinimų, grupuotų pagal šalies lauką

count

Grąžina atitinkančių geoobjektų skaičių.

Sintaksė	count(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `count("stotys",group_by:="šalis")` → stočių skaičius, grupuotas pagal šalių lauką

count_distinct

Grąžina skirtingų reikšmių skaičių.

Sintaksė	count_distinct(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `count_distinct("stotys",group_by:="šalis")` → skaičius skirtingų stočių reikšmių, grupuotų pagal šalies lauką

count_missing

Grąžina trūkstamų (NULL) reikšmių skaičių.

Sintaksė	count_missing(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `count_missing("stotys",group_by:="šalis")` → skaičius trūkstamų (NULL) stočių reikšmių, grupuotų pagal šalies lauką

iqr

Grąžina pagal lauką ar išraišką paskaičiuotą tarpkvartilinį diapazoną.

Sintaksė	iqr(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `iqr("gyventojai",group_by:="valstija")` → tarpkvartilinis gyventojų reikšmės diapazonas sugrupuotas pagal valstijos lauką

majority

Grąžina lauko ar išraiškos agreguotą reikšmių daugumą (dažniausiai pasitaikančią reikšmę).

Sintaksė	majority(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `majority("klasė",group_by:="valstija")` → dažniausiai pasitaikanti klasės reikšmė sugrupavus pagal valstijos lauką

max_length

Grąžina maksimalų lauko ar išraiškos teksto ilgį.

Sintaksė	max_length(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `max_length("miesto_pavadinimas",group_by:="valstybe")` → maksimalus miesto_pavadinimas ilgis grupuojant pagal lauką valstybe

maximum

Grąžina agreguotą lauko ar išraiškos reikšmės maksimalią reikšmę.

Sintaksė	maximum(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `maximum("gyventojai",group_by:="valstybe")` → maksimali gyventojų reikšmė grupuojant pagal lauką valstybe

mean

Grąžina lauko ar išraiškos reikšmės agreguotą vidurkį.

Sintaksė	mean(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `mean("gyventojai",group_by:="vaklstybe")` → vidutinė gyventojų skaičiaus reikšmė grupuojant pagal valstybės lauką

median

Grąžina agreguotą lauko ar išraiškos reikšmės medianą.

Sintaksė	median(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `median("gyventojai",group_by:="vaklstybe")` → gyventojų skaičiaus medianos reikšmė grupuojant pagal valstybės lauką

min_length

Grąžina minimalų lauko ar išraiškos teksto ilgį.

Sintaksė	min_length(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `min_length("miesto_pavadinimas",group_by:="valstybe")` → minimalus miesto_pavadinimas ilgis grupuojant pagal lauką valstybe

minimum

Grąžina agreguotą minimalią lauko ar išraiškos reikšmės reikšmę.

Sintaksė	minimum(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `minimum("gyventojai",group_by:="valstybe")` → minimali gyventojų reikšmė grupuojant pagal lauką valstybe

minority

Grąžina lauko ar išraiškos agreguotą reikšmių mažumą (rečiausiai pasitaikančią reikšmę).

Sintaksė	minority(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `minority("klasė",group_by:="valstija")` → rečiausiai pasitaikanti klasės reikšmė sugrupavus pagal valstijos lauką

q1

Grąžina pagal lauką ar išraišką paskaičiuotą pirmo kvartilio diapazoną.

Sintaksė	q1(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `q1("gyventojai",group_by:="valstija")` → pirmo kvartilio gyventojų reikšmių diapazonas sugrupuotas pagal valstijos lauką

q3

Grąžina pagal lauką ar išraišką paskaičiuotą trečio kvartilio diapazoną.

Sintaksė	q3(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `q3("gyventojai",group_by:="valstija")` → trečio kvartilio gyventojų reikšmių diapazonas sugrupuotas pagal valstijos lauką

range

Returns the aggregate range of values (maximum - minimum) from a field or expression.

Sintaksė	range(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `range("population",group_by:="state")` → range of population values, grouped by state field

relation_aggregate

Returns an aggregate value calculated using all matching child features from a layer relation.

Sintaksė	relation_aggregate(relation, aggregate, expression, [concatenator=''], [order_by]) [] žymi neprivalomus argumentus
Argumentai	• **relation** - a string, representing a relation ID • **aggregate** - tekstas, nurodantis, kokį skaičiuoti agregatą. Galimos parinktys yra: – count – count_distinct – count_missing – minimum ar min – maximum ar max – sum – mean – median – stdev – stdevsample – range – minority – majority – q1: pirmas kvartilis – q3: trečias kvartilis – iqr: tarpkvartilinis diapazonas – min_length: minimalus teksto ilgis – max_length: maksimalus teksto ilgis – concatenate: sujungti tekstus jungtuku – concatenate_unique: sujungti unikalius tekstus jungtuku – collect: kurti agreguotą kelių dalių geometriją – array_agg: kurti agreguotų reikšmių masyvą • **expression** - vidinė išraiška arba lauko pavadinimas agregavimui • **concatenator** - optional string to use to join values for ‚concatenate' aggregate • **order_by** - optional expression to order the features used for calculating the aggregate. Fields and geometry are from the features on the joined layer. By default, the features will be returned in an unspecified order.
Pavyzdžiai	• `relation_aggregate(relation:='my_relation', aggregate:='mean',expression:="passengers")` → mean value of all matching child features using the ‚my_relation' relation • `relation_aggregate('my_relation','sum', "passengers"/7)` → sum of the passengers field divided by 7 for all matching child features using the ‚my_relation' relation • `relation_aggregate('my_relation','concatenate', "towns", concatenator:=',')` → comma separated list of the towns field for all matching child features using the ‚my_relation' relation • `relation_aggregate('my_relation','array_agg', "id")` → array of the id field from all matching child features using the ‚my_relation' relation

Further reading: *Setting relations between multiple layers*

stdev

Returns the aggregate standard deviation value from a field or expression.

Sintaksė	stdev(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `stdev("population",group_by:="state")` → standard deviation of population value, grouped by state field

sum

Returns the aggregate summed value from a field or expression.

Sintaksė	sum(expression, [group_by], [filter]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - agreguojamo lauko subišraiška • **group_by** - neprivaloma išraiška, kuri naudojama agregatų grupavimui • **filter** - neprivaloma išraiška, kuri naudojama skaičiuojamo agregato geoobjektų filtravimui
Pavyzdžiai	• `sum("population",group_by:="state")` → summed population value, grouped by state field

9.2.2 Array Functions

This group contains functions to create and manipulate arrays (also known as list data structures). The order of values within the array matters, unlike the *‚map' data structure*, where the order of key-value pairs is irrelevant and values are identified by their keys.

array

Returns an array containing all the values passed as parameter.

Sintaksė	array(value1, value2, …)
Argumentai	• **value** - a value
Pavyzdžiai	• `array(2,10)` → [2, 10] • `array(2,10)[0]` → 2

array_all

Returns TRUE if an array contains all the values of a given array.

Sintaksė	array_all(array_a, array_b)
Argumentai	• **array_a** - an array • **array_b** - the array of values to search
Pavyzdžiai	• `array_all(array(1,2,3),array(2,3))` → TRUE • `array_all(array(1,2,3),array(1,2,4))` → FALSE

array_append

Returns an array with the given value added at the end.

Sintaksė	array_append(array, value)
Argumentai	• **array** - an array • **value** - the value to add
Pavyzdžiai	• `array_append(array(1,2,3),4)` → [1, 2, 3, 4]

array_cat

Returns an array containing all the given arrays concatenated.

Sintaksė	array_cat(array1, array2, …)
Argumentai	• **array** - an array
Pavyzdžiai	• `array_cat(array(1,2),array(2,3))` → [1, 2, 2, 3]

array_contains

Returns TRUE if an array contains the given value.

Sintaksė	array_contains(array, value)
Argumentai	• **array** - an array • **value** - the value to search
Pavyzdžiai	• `array_contains(array(1,2,3),2)` → TRUE

array_count

Counts the number of occurrences of a given value in an array.

Sintaksė	array_count(array, value)
Argumentai	• **array** - an array • **value** - the value to count
Pavyzdžiai	• `array_count(array('a', 'b', 'c', 'b'), 'b')` → 2

array_distinct

Returns an array containing distinct values of the given array.

Sintaksė	array_distinct(array)
Argumentai	• **array** - an array
Pavyzdžiai	• `array_distinct(array(1,2,3,2,1))` → [1, 2, 3]

array_filter

Returns an array with only the items for which the expression evaluates to true.

Sintaksė	array_filter(array, expression, [limit=0]) [] žymi neprivalomus argumentus
Argumentai	• **array** - an array • **expression** - an expression to evaluate on each item. The variable *@element* will be replaced by the current value. • **limit** - maximum number of elements to be returned. Use 0 to return all values.
Pavyzdžiai	• `array_filter(array(1,2,3),@element < 3)` → [1, 2] • `array_filter(array(1,2,3),@element < 3, 1)` → [1]

array_find

Returns the lowest index (0 for the first one) of a value within an array. Returns -1 if the value is not found.

Sintaksė	array_find(array, value)
Argumentai	• **array** - an array • **value** - the value to search
Pavyzdžiai	• `array_find(array('a', 'b', 'c'), 'b')` → 1 • `array_find(array('a', 'b', 'c', 'b'), 'b')` → 1

array_first

Returns the first value of an array.

Sintaksė	array_first(array)
Argumentai	• **array** - an array
Pavyzdžiai	• `array_first(array('a','b','c'))` → ‚a‘

array_foreach

Returns an array with the given expression evaluated on each item.

Sintaksė	array_foreach(array, expression)
Argumentai	• **array** - an array • **expression** - an expression to evaluate on each item. The variable *@element* will be replaced by the current value and the variable *@counter* by the current index (starting with 0).
Pavyzdžiai	• `array_foreach(array('a','b','c'),upper(@element))` → [‚A‘, ‚B‘, ‚C‘] • `array_foreach(array(1,2,3),@element + 10)` → [11, 12, 13] • `array_foreach(array(1,2,3),@element + @counter)` → [1, 3, 5]

array_get

Returns the Nth value (0 for the first one) or the last -Nth value (-1 for the last one) of an array.

Sintaksė	array_get(array, pos)
Argumentai	• **array** - an array • **pos** - the index to get (0 based)
Pavyzdžiai	• `array_get(array('a','b','c'),1)` → ‚b‘ • `array_get(array('a','b','c'),-1)` → ‚c‘

Patarimas

You can also use the *index operator ([])* to get a value from an array.

array_insert

Returns an array with the given value added at the given position.

Sintaksė	array_insert(array, pos, value)
Argumentai	• **array** - an array • **pos** - the position where to add (0 based) • **value** - the value to add
Pavyzdžiai	• `array_insert(array(1,2,3),1,100)` → [1, 100, 2, 3]

array_intersect

Returns TRUE if at least one element of array1 exists in array2.

Sintaksė	array_intersect(array1, array2)
Argumentai	• **array1** - an array • **array2** - another array
Pavyzdžiai	• `array_intersect(array(1,2,3,4),array(4,0,2,5))` → TRUE

array_last

Returns the last value of an array.

Sintaksė	array_last(array)
Argumentai	• **array** - an array
Pavyzdžiai	• `array_last(array('a','b','c'))` → ‚c‘

array_length

Returns the number of elements of an array.

Sintaksė	array_length(array)
Argumentai	• **array** - an array
Pavyzdžiai	• `array_length(array(1,2,3))` → 3

array_majority

Returns the most common values in an array.

Sintaksė	array_majority(array, [option='all']) [] žymi neprivalomus argumentus
Argumentai	• **array** - an array • **option='all'** - a string specifying the return values handling. Valid options are: – all: Default, all most common values are returned in an array. – any: Returns one of the most common values. – median: Returns the median of the most common values. Non arithmetic values are ignored. – real_majority: Returns the value which occurs more than half the size of the array.
Pavyzdžiai	• `array_majority(array(0,1,42,42,43), 'all')` → [42] • `array_majority(array(0,1,42,42,43,1), 'all')` → [42, 1] • `array_majority(array(0,1,42,42,43,1), 'any')` → 1 or 42 • `array_majority(array(0,1,1,2,2), 'median')` → 1.5 • `array_majority(array(0,1,42,42,43), 'real_majority')` → NULL • `array_majority(array(0,1,42,42,43,42), 'real_majority')` → NULL • `array_majority(array(0,1,42,42,43,42,42), 'real_majority')` → 42

array_max

Returns the maximum value of an array.

Sintaksė	array_max(array)
Argumentai	• **array** - an array
Pavyzdžiai	• `array_max(array(0,42,4,2))` → 42

array_mean

Returns the mean of arithmetic values in an array. Non numeric values in the array are ignored.

Sintaksė	array_mean(array)
Argumentai	• **array** - an array
Pavyzdžiai	• `array_mean(array(0,1,7,66.6,135.4))` → 42 • `array_mean(array(0,84,'a','b','c'))` → 42

array_median

Returns the median of arithmetic values in an array. Non arithmetic values in the array are ignored.

Sintaksė	array_median(array)
Argumentai	• **array** - an array
Pavyzdžiai	• `array_median(array(0,1,42,42,43))` → 42 • `array_median(array(0,1,2,42,'a','b'))` → 1.5

array_min

Returns the minimum value of an array.

Sintaksė	array_min(array)
Argumentai	• **array** - an array
Pavyzdžiai	• `array_min(array(43,42,54))` → 42

array_minority

Returns the less common values in an array.

Sintaksė	array_minority(array, [option='all']) [] žymi neprivalomus argumentus
Argumentai	• **array** - an array • **option='all'** - a string specifying the return values handling. Valid options are: – all: Default, all less common values are returned in an array. – any: Returns one of the less common values. – median: Returns the median of the less common values. Non arithmetic values are ignored. – real_minority: Returns values which occur less than half the size of the array.
Pavyzdžiai	• `array_minority(array(0,42,42), 'all')` → [0] • `array_minority(array(0,1,42,42), 'all')` → [0, 1] • `array_minority(array(0,1,42,42,43,1), 'any')` → 0 or 43 • `array_minority(array(1,2,3,3), 'median')` → 1.5 • `array_minority(array(0,1,42,42,43), 'real_minority')` → [42, 43, 0, 1] • `array_minority(array(0,1,42,42,43,42), 'real_minority')` → [42, 43, 0, 1] • `array_minority(array(0,1,42,42,43,42,42), 'real_minority')` → [43, 0, 1]

array_prepend

Returns an array with the given value added at the beginning.

Sintaksė	array_prepend(array, value)
Argumentai	• **array** - an array • **value** - the value to add
Pavyzdžiai	• `array_prepend(array(1,2,3),0)` → [0, 1, 2, 3]

array_prioritize

Returns an array sorted using the ordering specified in another array. Values which are present in the first array but are missing from the second array will be added to the end of the result.

Sintaksė	array_prioritize(array, array_prioritize)
Argumentai	• **array** - an array • **array_prioritize** - an array with values ordered by priority
Pavyzdžiai	• `array_prioritize(array(1, 8, 2, 5), array(5, 4, 2, 1, 3, 8))` → [5, 2, 1, 8] • `array_prioritize(array(5, 4, 2, 1, 3, 8), array(1, 8, 6, 5))` → [1, 8, 5, 4, 2, 3]

array_remove_all

Returns an array with all the entries of the given value removed.

Sintaksė	array_remove_all(array, value)
Argumentai	• **array** - an array • **value** - the values to remove
Pavyzdžiai	• `array_remove_all(array('a','b','c','b'),'b')` → [‚a‘, ‚c‘]

array_remove_at

Returns an array with the item at the given index removed. Supports positive (0 for the first element) and negative (the last -Nth value, -1 for the last element) index.

Sintaksė	array_remove_at(array, pos)
Argumentai	• **array** - an array • **pos** - the position to remove (0 based)
Pavyzdžiai	• `array_remove_at(array(1, 2, 3), 1)` → [1, 3] • `array_remove_at(array(1, 2, 3), -1)` → [1, 2]

array_replace

Returns an array with the supplied value, array, or map of values replaced.

Value & array variant

Returns an array with the supplied value or array of values replaced by another value or an array of values.

Sintaksė	array_replace(array, before, after)
Argumentai	• **array** - the input array • **before** - the value or array of values to replace • **after** - the value or array of values to use as a replacement
Pavyzdžiai	• `array_replace(array('QGIS','SHOULD','ROCK'),'SHOULD', 'DOES')` → [‚QGIS‘, ‚DOES‘, ‚ROCK‘] • `array_replace(array(3,2,1),array(1,2,3),array(7,8,9))` → [9, 8, 7] • `array_replace(array('Q','G','I','S'),array('Q','S'),'-')` → [‚-‘, ‚G‘, ‚I‘, ‚-‘]

Map variant

Returns an array with the supplied map keys replaced by their paired values.

Sintaksė	array_replace(array, map)
Argumentai	• **array** - the input array • **map** - the map containing keys and values
Pavyzdžiai	• `array_replace(array('APP', 'SHOULD', 'ROCK'),map('APP', 'QGIS','SHOULD','DOES'))` → [‚QGIS‘, ‚DOES‘, ‚ROCK‘]

Further reading: *replace*, *regexp_replace*

array_reverse

Returns the given array with array values in reversed order.

Sintaksė	array_reverse(array)
Argumentai	• **array** - an array
Pavyzdžiai	• `array_reverse(array(2,4,0,10))` → [10, 0, 4, 2]

array_slice

Returns a portion of the array. The slice is defined by the start_pos and end_pos arguments.

Sintaksė	array_slice(array, start_pos, end_pos)
Argumentai	• **array** - an array • **start_pos** - the index of the start position of the slice (0 based). The start_pos index is included in the slice. If you use a negative start_pos, the index is counted from the end of the list (-1 based). • **end_pos** - the index of the end position of the slice (0 based). The end_pos index is included in the slice. If you use a negative end_pos, the index is counted from the end of the list (-1 based).
Pavyzdžiai	• `array_slice(array(1,2,3,4,5),0,3)` → [1, 2, 3, 4] • `array_slice(array(1,2,3,4,5),0,-1)` → [1, 2, 3, 4, 5] • `array_slice(array(1,2,3,4,5),-5,-1)` → [1, 2, 3, 4, 5] • `array_slice(array(1,2,3,4,5),0,0)` → [1] • `array_slice(array(1,2,3,4,5),-2,-1)` → [4, 5] • `array_slice(array(1,2,3,4,5),-1,-1)` → [5] • `array_slice(array('Dufour','Valmiera','Chugiak', 'Brighton'),1,2)` → [‚Valmiera‘, ‚Chugiak‘] • `array_slice(array('Dufour','Valmiera','Chugiak', 'Brighton'),-2,-1)` → [‚Chugiak‘, ‚Brighton‘]

array_sort

Returns the provided array with its elements sorted.

Sintaksė	array_sort(array, [ascending=true]) [] žymi neprivalomus argumentus
Argumentai	• **array** - an array • **ascending** - set this parameter to false to sort the array in descending order
Pavyzdžiai	• `array_sort(array(3,2,1))` → [1, 2, 3]

array_sum

Returns the sum of arithmetic values in an array. Non numeric values in the array are ignored.

Sintaksė	array_sum(array)
Argumentai	• **array** - an array
Pavyzdžiai	• `array_sum(array(0,1,39.4,1.6,'a'))` → 42.0

array_to_string

Concatenates array elements into a string separated by a delimiter and using optional string for empty values.

Sintaksė	array_to_string(array, [delimiter=‘,‘], [empty_value=‘‘]) [] žymi neprivalomus argumentus
Argumentai	• **array** - the input array • **delimiter** - the string delimiter used to separate concatenated array elements • **empty_value** - the optional string to use as replacement for empty (zero length) matches
Pavyzdžiai	• `array_to_string(array('1','2','3'))` → ‚1,2,3‘ • `array_to_string(array(1,2,3),'-')` → ‚1-2-3‘ • `array_to_string(array('1','','3'),',','0')` → ‚1,0,3‘

Further reading: *string_to_array*

generate_series

Creates an array containing a sequence of numbers.

Sintaksė	generate_series(start, stop, [step=1]) [] žymi neprivalomus argumentus
Argumentai	• **start** - first value of the sequence • **stop** - value that ends the sequence once reached • **step** - value used as the increment between values
Pavyzdžiai	• `generate_series(1,5)` → [1, 2, 3, 4, 5] • `generate_series(5,1,-1)` → [5, 4, 3, 2, 1]

geometries_to_array

Splits a geometry into simpler geometries in an array.

Sintaksė	geometries_to_array(geometry)
Argumentai	• **geometry** - the input geometry
Pavyzdžiai	• `geometries_to_array(geom_from_wkt('MultiPoint (1 2, 5 21)'))` → An array containing ‚Point (1 2)‘ and ‚Point (5 21)‘ • `geometries_to_array(geom_from_wkt('GeometryCollection (Polygon ((5 8, 4 1, 3 2, 5 8)),LineString (3 2, 4 2))'))` → an array of a polygon and a line geometries • `geom_to_wkt(geometries_to_array(geom_from_wkt('GeometryCollection (Polygon ((5 8, 4 1, 3 2, 5 8)),LineString (3 2, 4 2))'))[0])` → ‚Polygon ((5 8, 4 1, 3 2, 5 8))‘ • `geometries_to_array(geom_from_wkt('MULTIPOLYGON(((5 5,0 0, 0 10,5 5)),((5 5,10 10,10 0,5 5))'))` → an array of two polygon geometries

regexp_matches

Returns an array of all strings captured by capturing groups, in the order the groups themselves appear in the supplied regular expression against a string.

Sintaksė	regexp_matches(string, regex, [empty_value=‘‘]) [] žymi neprivalomus argumentus
Argumentai	• **string** - the string to capture groups from against the regular expression • **regex** - the regular expression used to capture groups • **empty_value** - the optional string to use as replacement for empty (zero length) matches
Pavyzdžiai	• `regexp_matches('QGIS=>rocks','(.*)=>(.*)')` → [‚QGIS‘, ‚rocks‘] • `regexp_matches('key=>','(.*)=>(.*)','empty value')` → [‚key‘, ‚empty value‘]

Further reading: *substr*, *regexp_substr*

string_to_array

Splits string into an array using supplied delimiter and optional string for empty values.

Sintaksė	string_to_array(string, [delimiter=‘,‘], [empty_value=‘‘]) [] žymi neprivalomus argumentus
Argumentai	• **string** - the input string • **delimiter** - the string delimiter used to split the input string • **empty_value** - the optional string to use as replacement for empty (zero length) matches
Pavyzdžiai	• `string_to_array('1,2,3',',')` → [‚1‘, ‚2‘, ‚3‘] • `string_to_array('1,,3',',','0')` → [‚1‘, ‚0‘, ‚3‘]

Further reading: *array_to_string*

9.2.3 Color Functions

This group contains functions for manipulating colors.

color_cmyk

Returns a string representation of a color based on its cyan, magenta, yellow and black components

Sintaksė	color_cmyk(cyan, magenta, yellow, black)
Argumentai	• **cyan** - cyan component of the color, as a percentage integer value from 0 to 100 • **magenta** - magenta component of the color, as a percentage integer value from 0 to 100 • **yellow** - yellow component of the color, as a percentage integer value from 0 to 100 • **black** - black component of the color, as a percentage integer value from 0 to 100
Pavyzdžiai	• `color_cmyk(100,50,0,10)` → ‚0,115,230‘

color_cmyka

Returns a string representation of a color based on its cyan, magenta, yellow, black and alpha (transparency) components

Sintaksė	color_cmyka(cyan, magenta, yellow, black, alpha)
Argumentai	• **cyan** - cyan component of the color, as a percentage integer value from 0 to 100 • **magenta** - magenta component of the color, as a percentage integer value from 0 to 100 • **yellow** - yellow component of the color, as a percentage integer value from 0 to 100 • **black** - black component of the color, as a percentage integer value from 0 to 100 • **alpha** - alpha component as an integer value from 0 (completely transparent) to 255 (opaque).
Pavyzdžiai	• `color_cmyka(100,50,0,10,200)` → ‚0,115,230,200‘

color_cmykf

Returns a color object based on its cyan, magenta, yellow, black and alpha components.

Sintaksė	color_cmykf(cyan, magenta, yellow, black, [alpha=1.0]) [] žymi neprivalomus argumentus
Argumentai	• **cyan** - cyan component as a float value from 0.0 to 1.0 • **magenta** - magenta component as a float value from 0.0 to 1.0 • **yellow** - yellow component as a float value from 0.0 to 1.0 • **black** - black component as a float value from 0.0 to 1.0 • **alpha** - alpha component as a float value from 0.0 to 1.0
Pavyzdžiai	• `color_cmykf(1,0.9,0.81,0.62)` → CMYKA: 1.00,0.90,0.81,0.62,1.00

color_grayscale_average

Applies a grayscale filter to a color and returns it. Returned type is the same as color argument, i.e. a color string representation or a color object.

Sintaksė	color_grayscale_average(color)
Argumentai	• **color** - a color string representation or a color object
Pavyzdžiai	• `color_grayscale_average('255,100,50')` → ‚135,135,135,255‘ • `color_grayscale_average(color_cmykf(0.6,0.5,0.1,0.8))` → CMYKA: 0.40,0.40,0.40,0.80,1.00

color_hsl

Returns a string representation of a color based on its hue, saturation, and lightness attributes.

Sintaksė	color_hsl(hue, saturation, lightness)
Argumentai	• **hue** - hue of the color, as an integer value from 0 to 360 • **saturation** - saturation percentage of the color as an integer value from 0 to 100 • **lightness** - lightness percentage of the color as an integer value from 0 to 100
Pavyzdžiai	• `color_hsl(100,50,70)` → ‚166,217,140‘

color_hsla

Returns a string representation of a color based on its hue, saturation, lightness and alpha (transparency) attributes

Sintaksė	color_hsla(hue, saturation, lightness, alpha)
Argumentai	• **hue** - hue of the color, as an integer value from 0 to 360 • **saturation** - saturation percentage of the color as an integer value from 0 to 100 • **lightness** - lightness percentage of the color as an integer value from 0 to 100 • **alpha** - alpha component as an integer value from 0 (completely transparent) to 255 (opaque).
Pavyzdžiai	• `color_hsla(100,50,70,200)` → ‚166,217,140,200‘

color_hslf

Returns a color object based on its hue, saturation, and lightness attributes.

Sintaksė	color_hslf(hue, saturation, lightness, [alpha=1.0]) [] žymi neprivalomus argumentus
Argumentai	• **hue** - hue of the color, as a float value from 0.0 to 1.0 • **saturation** - saturation of the color as a float value from 0.0 to 1.0 • **lightness** - lightness of the color as a float value from 0.0 to 1.0 • **alpha** - alpha component as a float value from 0.0 to 1.0
Pavyzdžiai	• `color_hslf(0.3,0.52,0.7)` → HSLA: 0.30,0.52,0.70,1.00

color_hsv

Returns a string representation of a color based on its hue, saturation, and value attributes.

Sintaksė	color_hsv(hue, saturation, value)
Argumentai	• **hue** - hue of the color, as an integer value from 0 to 360 • **saturation** - saturation percentage of the color as an integer value from 0 to 100 • **value** - value percentage of the color as an integer from 0 to 100
Pavyzdžiai	• `color_hsv(40,100,100)` → ‚255,170,0‘

color_hsva

Returns a string representation of a color based on its hue, saturation, value and alpha (transparency) attributes.

Sintaksė	color_hsva(hue, saturation, value, alpha)
Argumentai	• **hue** - hue of the color, as an integer value from 0 to 360 • **saturation** - saturation percentage of the color as an integer value from 0 to 100 • **value** - value percentage of the color as an integer from 0 to 100 • **alpha** - alpha component as an integer value from 0 (completely transparent) to 255 (opaque)
Pavyzdžiai	• `color_hsva(40,100,100,200)` → ‚255,170,0,200‘

color_hsvf

Returns a color object based on its hue, saturation, and value attributes.

Sintaksė	color_hsvf(hue, saturation, value, [alpha=1.0]) [] žymi neprivalomus argumentus
Argumentai	• **hue** - hue of the color, as a float value from 0.0 to 1.0 • **saturation** - saturation of the color as a float value from 0.0 to 1.0 • **value** - value of the color as as a float value from 0.0 to 1.0 • **alpha** - alpha component as a float value from 0.0 to 1.0
Pavyzdžiai	• `color_hsvf(0.4,1,0.6)` → HSVA: 0.40,1.00,0.60,1.00

color_mix

Returns a color mixing the red, green, blue, and alpha values of two provided colors based on a given ratio. Returned type is the same as color arguments, i.e. a color string representation or a color object.

Sintaksė	color_mix(color1, color2, ratio)
Argumentai	• **color1** - a color string or a color object • **color2** - a color string or a color object • **ratio** - a ratio
Pavyzdžiai	• `color_mix_rgb('0,0,0','255,255,255',0.5)` → ‚127,127,127,255‘ • `color_mix(color_cmykf(0.9,0.9,0.9,0.9),color_cmykf(0.1,0.1,0.1,0.1),0.5)` → CMYKA: 0.50,0.50,0.50,0.50,1.00

color_mix_rgb

Returns a string representing a color mixing the red, green, blue, and alpha values of two provided colors based on a given ratio.

Sintaksė	color_mix_rgb(color1, color2, ratio)
Argumentai	• **color1** - a color string • **color2** - a color string • **ratio** - a ratio
Pavyzdžiai	• `color_mix_rgb('0,0,0','255,255,255',0.5)` → ‚127,127,127,255‘

color_part

Returns a specific component from a color string or color object, e.g., the red component or alpha component.

Sintaksė	color_part(color, component)
Argumentai	• **color** - a color string or a color object • **component** - a string corresponding to the color component to return. Valid options are: – red: RGB red component (0-255) – green: RGB green component (0-255) – blue: RGB blue component (0-255) – alpha: alpha (transparency) value (0-255) – hue: HSV hue (0-360) – saturation: HSV saturation (0-100) – value: HSV value (0-100) – hsl_hue: HSL hue (0-360) – hsl_saturation: HSL saturation (0-100) – lightness: HSL lightness (0-100) – cyan: CMYK cyan component (0-100) – magenta: CMYK magenta component (0-100) – yellow: CMYK yellow component (0-100) – black: CMYK black component (0-100)
Pavyzdžiai	• `color_part('200,10,30','green')` → 10 • `to_int(color_part(color_cmykf(0.1,0.2,0.3,0.9),'black'))` → 90

color_rgb

Returns a string representation of a color based on its red, green, and blue components.

Sintaksė	color_rgb(red, green, blue)
Argumentai	• **red** - red component as an integer value from 0 to 255 • **green** - green component as an integer value from 0 to 255 • **blue** - blue component as an integer value from 0 to 255
Pavyzdžiai	• `color_rgb(255,127,0)` → ‚255,127,0‘

color_rgba

Returns a string representation of a color based on its red, green, blue, and alpha (transparency) components.

Sintaksė	color_rgba(red, green, blue, alpha)
Argumentai	• **red** - red component as an integer value from 0 to 255 • **green** - green component as an integer value from 0 to 255 • **blue** - blue component as an integer value from 0 to 255 • **alpha** - alpha component as an integer value from 0 (completely transparent) to 255 (opaque).
Pavyzdžiai	• `color_rgba(255,127,0,200)` → ‚255,127,0,200‘

color_rgbf

Returns a color object based on its red, green, blue and alpha components.

Sintaksė	color_rgbf(red, green, blue, [alpha=1.0]) [] žymi neprivalomus argumentus
Argumentai	• **red** - red component as a float value from 0.0 to 1.0 • **green** - green component as a float value from 0.0 to 1.0 • **blue** - blue component as a float value from 0.0 to 1.0 • **alpha** - alpha component as a float value from 0.0 to 1.0
Pavyzdžiai	• `color_rgbf(1.0,0.5,0)` → RGBA: 1.00,0.50,0.00,1.00

create_ramp

Returns a gradient ramp from a map of color strings and steps.

Sintaksė	create_ramp(map, [discrete=false]) [] žymi neprivalomus argumentus
Argumentai	• **map** - a map of color strings and steps • **discrete** - set this parameter to true to create a discrete color ramp
Pavyzdžiai	• `ramp_color(create_ramp(map(0,'0,0,0',1,'255,0,0')),1)` → ‚255,0,0,255‘

darker

Returns a darker (or lighter) color. Returned type is the same as color arguments, i.e. a color string representation or a color object.

Sintaksė	darker(color, factor)
Argumentai	• **color** - a color string or a color object • **factor** - an integer corresponding to the darkening factor: – if the factor is greater than 100, this function returns a darker color (e.g., setting factor to 200 returns a color that is half the brightness); – if the factor is less than 100, the return color is lighter, but using the lighter() function for this purpose is recommended; – if the factor is 0 or negative, the return value is unspecified.
Pavyzdžiai	• `darker('200,10,30', 200)` → ‚100,5,15,255‘

Further reading: *lighter*

lighter

Returns a lighter (or darker) color. Returned type is the same as color arguments, i.e. a color string representation or a color object.

Sintaksė	lighter(color, factor)
Argumentai	• **color** - a color string or a color object • **factor** - an integer corresponding to the lightening factor: – if the factor is greater than 100, this function returns a lighter color (e.g., setting factor to 150 returns a color that is 50% brighter); – if the factor is less than 100, the return color is darker, but using the darker() function for this purpose is recommended; – if the factor is 0 or negative, the return value is unspecified.
Pavyzdžiai	• `lighter('200,10,30', 200)` → ‚255,158,168,255‘

Further reading: *darker*

project_color

Returns a color from the project's color scheme.

Sintaksė	project_color(name)
Argumentai	• **name** - a color name
Pavyzdžiai	• `project_color('Logo color')` → ‚20,140,50‘

Further reading: *setting project colors*

project_color_object

Returns a color from the project's color scheme. Contrary to project_color which returns a color string representation, project_color_object returns a color object.

Sintaksė	project_color_object(name)
Argumentai	• **name** - a color name
Pavyzdžiai	• `project_color_object('Logo color')` → RGBA: 0.08,0.55,0.20,1.00

ramp_color

Returns a string representing a color from a color ramp.

Saved ramp variant

Returns a string representing a color from a saved ramp

Sintaksė	ramp_color(ramp_name, value)
Argumentai	• **ramp_name** - the name of the color ramp as a string, for example ‚Spectral‘ • **value** - the position on the ramp to select the color from as a real number between 0 and 1
Pavyzdžiai	• `ramp_color('Spectral',0.3)` → ‚253,190,115,255‘

Pastaba

The color ramps available vary between QGIS installations. This function may not give the expected results if you move your QGIS project between installations.

Expression-created ramp variant

Returns a string representing a color from an expression-created ramp

Sintaksė	ramp_color(ramp, value)
Argumentai	• **ramp** - the color ramp • **value** - the position on the ramp to select the color from as a real number between 0 and 1
Pavyzdžiai	• `ramp_color(create_ramp(map(0,'0,0,0',1,'255,0,0')),1)` → ‚255,0,0,255‘

Further reading: *Spalvų rampos kūrimas*, *Iškrentantis spalvų rampos parinkimas*

ramp_color_object

Returns a color object from a color ramp. Contrary to ramp_color which returns a color string representation, ramp_color_object returns a color object.

Saved ramp variant

Returns a color object from a saved ramp

Sintaksė	ramp_color_object(ramp_name, value)
Argumentai	• **ramp_name** - the name of the color ramp as a string, for example ‚Spectral‘ • **value** - the position on the ramp to select the color from as a real number between 0 and 1
Pavyzdžiai	• `ramp_color_object('Spectral',0.3)` → RGBA: 0.99,0.75,0.45,1.00

Pastaba

The color ramps available vary between QGIS installations. This function may not give the expected results if you move your QGIS project between installations.

Expression-created ramp variant

Returns a color object from an expression-created ramp

Sintaksė	ramp_color_object(ramp, value)
Argumentai	• **ramp** - the color ramp • **value** - the position on the ramp to select the color from as a real number between 0 and 1
Pavyzdžiai	• `ramp_color_object(create_ramp(map(0,color_rgbf(0,0,0),1, color_rgbf(1,0,0))),1)` → RGBA: 1.00,0.00,0.00,1.00

set_color_part

Sets a specific color component for a color string or a color object, e.g., the red component or alpha component.

Sintaksė	set_color_part(color, component, value)
Argumentai	• **color** - a color string or a color object • **component** - a string corresponding to the color component to set. Valid options are: – red: RGB red component (0-255) – green: RGB green component (0-255) – blue: RGB blue component (0-255) – alpha: alpha (transparency) value (0-255) – hue: HSV hue (0-360) – saturation: HSV saturation (0-100) – value: HSV value (0-100) – hsl_hue: HSL hue (0-360) – hsl_saturation: HSL saturation (0-100) – lightness: HSL lightness (0-100) – cyan: CMYK cyan component (0-100) – magenta: CMYK magenta component (0-100) – yellow: CMYK yellow component (0-100) – black: CMYK black component (0-100) • **value** - new value for color component, respecting the ranges listed above
Pavyzdžiai	• `set_color_part('200,10,30','green',50)` → ‚200,50,30,255‘ • `set_color_part(color_cmykf(0.21,0,0.92,0.70),'black',100)` → CMYKA: 0.21,0.00,0.92,1.00,1.00

9.2.4 Conditional Functions

This group contains functions to handle conditional checks in expressions.

CASE

CASE is used to evaluate a series of conditions and return a result for the first condition met. The conditions are evaluated sequentially, and if a condition is true, the evaluation stops, and the corresponding result is returned. If none of the conditions are true, the value in the ELSE clause is returned. Furthermore, if no ELSE clause is set and none of the conditions are met, NULL is returned.

CASE

WHEN *condition* THEN *result*

[…n]

[ELSE *result*]

END

[ ] marks optional components

Argumentai	• **WHEN condition** - A condition expression to evaluate • **THEN result** - If *condition* evaluates to True then *result* is evaluated and returned. • **ELSE result** - If none of the above conditions evaluated to True then *result* is evaluated and returned.
Pavyzdžiai	• `CASE WHEN "name" IS NULL THEN 'None' END` → Returns the string ‚None‘ if the „name“ field is NULL • `CASE WHEN $area > 10000 THEN 'Big property' WHEN $area > 5000 THEN 'Medium property' ELSE 'Small property' END` → Returns the string ‚Big property‘ if the area is bigger than 10000, ‚Medium property‘ if the area is between 5000 and 10000, and ‚Small property‘ for others

coalesce

Returns the first non-NULL value from the expression list.

This function can take any number of arguments.

Sintaksė	coalesce(expression1, expression2, …)
Argumentai	• **expression** - any valid expression or value, regardless of type.
Pavyzdžiai	• `coalesce(NULL, 2)` → 2 • `coalesce(NULL, 2, 3)` → 2 • `coalesce(7, NULL, 3*2)` → 7 • `coalesce("fieldA", "fallbackField", 'ERROR')` → value of fieldA if it is non-NULL else the value of „fallbackField“ or the string ‚ERROR‘ if both are NULL

if

Tests a condition and returns a different result depending on the conditional check.

Sintaksė	if(condition, result_when_true, result_when_false)
Argumentai	• **condition** - the condition which should be checked • **result_when_true** - the result which will be returned when the condition is true or another value that does not convert to false. • **result_when_false** - the result which will be returned when the condition is false or another value that converts to false like 0 or ‚‘. NULL will also be converted to false.
Pavyzdžiai	• `if( 1+1=2, 'Yes', 'No' )` → ‚Yes‘ • `if( 1+1=3, 'Yes', 'No' )` → ‚No‘ • `if( 5 > 3, 1, 0)` → 1 • `if( '', 'It is true (not empty)', 'It is false (empty)' )` → ‚It is false (empty)‘ • `if( ' ', 'It is true (not empty)', 'It is false (empty)' )` → ‚It is true (not empty)‘ • `if( 0, 'One', 'Zero' )` → ‚Zero‘ • `if( 10, 'One', 'Zero' )` → ‚One‘

nullif

Returns a NULL value if value1 equals value2; otherwise it returns value1. This can be used to conditionally substitute values with NULL.

Sintaksė	nullif(value1, value2)
Argumentai	• **value1** - The value that should either be used or substituted with NULL. • **value2** - The control value that will trigger the NULL substitution.
Pavyzdžiai	• `nullif('(none)', '(none)')` → NULL • `nullif('text', '(none)')` → ‚text‘ • `nullif("name", '')` → NULL, if name is an empty string (or already NULL), the name in any other case.

regexp_match

Return the first matching position matching a regular expression within an unicode string, or 0 if the substring is not found.

Sintaksė	regexp_match(input_string, regex)
Argumentai	• **input_string** - the string to test against the regular expression • **regex** - The regular expression to test against. Backslash characters must be double escaped (e.g., „\\s“ to match a white space character or „\\b“ to match a word boundary).
Pavyzdžiai	• `regexp_match('QGIS ROCKS','\\sROCKS')` → 5 • `regexp_match('Budač','udač\\b')` → 2

try

Tries an expression and returns its value if error-free. If the expression returns an error, an alternative value will be returned when provided otherwise the function will return NULL.

Sintaksė	try(expression, [alternative]) [] žymi neprivalomus argumentus
Argumentai	• **expression** - the expression which should be run • **alternative** - the result which will be returned if the expression returns an error.
Pavyzdžiai	• `try( to_int( '1' ), 0 )` → 1 • `try( to_int( 'a' ), 0 )` → 0 • `try( to_date( 'invalid_date' ) )` → NULL

9.2.5 Conversions Functions

This group contains functions to convert one data type to another (e.g., string from/to integer, binary from/to string, string to date, …).

from_base64

Decodes a string in the Base64 encoding into a binary value.

Sintaksė	from_base64(string)
Argumentai	• **string** - the string to decode
Pavyzdžiai	• `from_base64('UUdJUw==')` → ‚QGIS‘

hash

Creates a hash from a string with a given method. One byte (8 bits) is represented with two hex ‚digits“, so ‚md4‘ (16 bytes) produces a 16 * 2 = 32 character long hex string and ‚keccak_512‘ (64 bytes) produces a 64 * 2 = 128 character long hex string.

Sintaksė	hash(string, method)
Argumentai	• **string** - the string to hash • **method** - The hash method among ‚md4‘, ‚md5‘, ‚sha1‘, ‚sha224‘, ‚sha384‘, ‚sha512‘, ‚sha3_224‘, ‚sha3_256‘, ‚sha3_384‘, ‚sha3_512‘, ‚keccak_224‘, ‚keccak_256‘, ‚keccak_384‘, ‚keccak_512‘
Pavyzdžiai	• `hash('QGIS', 'md4')` → ‚c0fc71c241cdebb6e888cbac0e2b68eb‘ • `hash('QGIS', 'md5')` → ‚57470aaa9e22adaefac7f5f342f1c6da‘ • `hash('QGIS', 'sha1')` → ‚f87cfb2b74cdd5867db913237024e7001e62b114‘ • `hash('QGIS', 'sha224')` → ‚4093a619ada631c770f44bc643ead18fb393b93d6a6af1861fcfece0‘ • `hash('QGIS', 'sha256')` → ‚eb045cba7a797aaa06ac58830846e40c8e8c780bc0676d3393605fae50c • `hash('QGIS', 'sha384')` → ‚91c1de038cc3d09fdd512e99f9dd9922efadc39ed21d3922e69a4305cc2 • `hash('QGIS', 'sha512')` → ‚c2c092f2ab743bf8edbeb6d028a745f30fc720408465ed369421f0a4e20 • `hash('QGIS', 'sha3_224')` → ‚467f49a5039e7280d5d42fd433e80d203439e338eaabd701f0d6c17d • `hash('QGIS', 'sha3_256')` → ‚540f7354b6b8a6e735f2845250f15f4f3ba4f666c55574d9e9354575 • `hash('QGIS', 'sha3_384')` → ‚96052da1e77679e9a65f60d7ead961b28797782314478638 6eb4364 • `hash('QGIS', 'sha3_512')` → ‚900d079dc69761da113980253aa8ac0414a8bd6d09879a916228f874 • `hash('QGIS', 'keccak_224')` → ‚5b0ce6acef8b0a121d4ac4f3eaa8503c799ad4e26a3392d1fb2014 • `hash('QGIS', 'keccak_256')` → ‚991c520aa6815392de24087f61b2ae0fd56abbfeee4a8ca019c101 • `hash('QGIS', 'keccak_384')` → ‚c57a3aed9d856fa04e5eeee9b62b6e027cca81ba574116d3cc1f0d4

md5

Creates a md5 hash from a string.

Sintaksė	md5(string)
Argumentai	• **string** - the string to hash
Pavyzdžiai	• `md5('QGIS')` → ‚57470aaa9e22adaefac7f5f342f1c6da‘

sha256

Creates a sha256 hash from a string.

Sintaksė	sha256(string)
Argumentai	• **string** - the string to hash
Pavyzdžiai	• `sha256('QGIS')` → ‚eb045cba7a797aaa06ac58830846e40c8e8c780bc0676d3393605fae50c05309‘

to_base64

Encodes a binary value into a string, using the Base64 encoding.

Sintaksė	to_base64(value)
Argumentai	• **value** - the binary value to encode
Pavyzdžiai	• `to_base64('QGIS')` → ‚UUdJUw==‘

to_bool

Converts a given value to a boolean. The function will return false if the value is NULL, an empty string, an empty list, or 0.

Sintaksė	to_bool(value)
Argumentai	• **value** - value to convert to boolean
Pavyzdžiai	• `to_bool('')` → false • `to_bool('123')` → true • `to_bool('false')` → true • `to_bool(0)` → false • `to_bool(1)` → true • `to_bool(null)` → false

to_date

Converts a string into a date object. An optional format string can be provided to parse the string; see QDate::fromString or the documentation of the format_date function for additional documentation on the format. By default the current QGIS user locale is used.

Sintaksė	to_date(string, [format], [language]) [] žymi neprivalomus argumentus
Argumentai	• **string** - string representing a date value • **format** - format used to convert the string into a date • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to convert the string into a date. By default the current QGIS user locale is used.
Pavyzdžiai	• `to_date('2012-05-04')` → 2012-05-04 • `to_date('June 29, 2019','MMMM d, yyyy')` → 2019-06-29, if the current locale uses the name ‚June' for the sixth month, otherwise an error occurs • `to_date('29 juin, 2019','d MMMM, yyyy','fr')` → 2019-06-29

Further reading: *format_date*

to_datetime

Converts a string into a datetime object. An optional format string can be provided to parse the string; see QDate::fromString, QTime::fromString or the documentation of the format_date function for additional documentation on the format. By default the current QGIS user locale is used.

Sintaksė	to_datetime(string, [format], [language]) [] žymi neprivalomus argumentus
Argumentai	• **string** - string representing a datetime value • **format** - format used to convert the string into a datetime • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to convert the string into a datetime. By default the current QGIS user locale is used.
Pavyzdžiai	• `to_datetime('2012-05-04 12:50:00')` → 2012-05-04T12:50:00 • `to_datetime('June 29, 2019 @ 12:34','MMMM d, yyyy @ HH:mm')` → 2019-06-29T12:34, if the current locale uses the name ‚June' for the sixth month, otherwise an error occurs • `to_datetime('29 juin, 2019 @ 12:34','d MMMM, yyyy @ HH:mm', 'fr')` → 2019-06-29T12:34

Further reading: *format_date*

to_decimal

Converts a degree, minute, second coordinate to its decimal equivalent.

Sintaksė	to_decimal(value)
Argumentai	• **value** - A degree, minute, second string.
Pavyzdžiai	• `to_decimal('6°21\'16.445')` → 6.3545680555

to_dm

Converts a coordinate to degree, minute.

Sintaksė	to_dm(coordinate, axis, precision, [formatting=]) [] žymi neprivalomus argumentus
Argumentai	• **coordinate** - A latitude or longitude value. • **axis** - The axis of the coordinate. Either ‚x' or ‚y'. • **precision** - Number of decimals. • **formatting** - Designates the formatting type. Acceptable values are NULL (default), ‚aligned' or ‚suffix'.
Pavyzdžiai	• `to_dm(6.1545681, 'x', 3)` → 6°9.274′ • `to_dm(6.1545681, 'y', 4, 'aligned')` → 6°09.2741′N • `to_dm(6.1545681, 'y', 4, 'suffix')` → 6°9.2741′N

to_dms

Converts a coordinate to degree, minute, second.

Sintaksė	to_dms(coordinate, axis, precision, [formatting=]) [] žymi neprivalomus argumentus
Argumentai	• **coordinate** - A latitude or longitude value. • **axis** - The axis of the coordinate. Either ‚x' or ‚y'. • **precision** - Number of decimals. • **formatting** - Designates the formatting type. Acceptable values are NULL (default), ‚aligned' or ‚suffix'.
Pavyzdžiai	• `to_dms(6.1545681, 'x', 3)` → 6°9′16.445″ • `to_dms(6.1545681, 'y', 4, 'aligned')` → 6°09′16.4452″N • `to_dms(6.1545681, 'y', 4, 'suffix')` → 6°9′16.4452″N

to_int

Converts a string to integer number. If a value cannot be converted to integer the expression is invalid (e.g ‚123asd' is invalid).

Sintaksė	to_int(string)
Argumentai	• **string** - string to convert to integer number
Pavyzdžiai	• `to_int('123')` → 123

to_interval

Converts a string to an interval type. Can be used to take days, hours, month, etc of a date.

Sintaksė	to_interval(string)
Argumentai	• **string** - a string representing an interval. Allowable formats include {n} days {n} hours {n} months.
Pavyzdžiai	• `to_interval('1 day 2 hours')` → interval: 1.08333 days • `to_interval( '0.5 hours' )` → interval: 30 minutes • `to_datetime('2012-05-05 12:00:00') - to_interval('1 day 2 hours')` → 2012-05-04T10:00:00

to_real

Converts a string to a real number. If a value cannot be converted to real the expression is invalid (e.g ‚123.56asd' is invalid). Numbers are rounded after saving changes if the precision is smaller than the result of the conversion.

Sintaksė	to_real(string)
Argumentai	• **string** - string to convert to real number
Pavyzdžiai	• `to_real('123.45')` → 123.45

to_string

Converts a number to string. The conversion is not locale-aware, see ‚format_number' for a locale-aware alternative.

Sintaksė	to_string(number)
Argumentai	• **number** - Integer or real value. The number to convert to string.
Pavyzdžiai	• `to_string(1.23)` → ‚1.23'

Further reading: *format_number*

to_time

Converts a string into a time object. An optional format string can be provided to parse the string; see QTime::fromString for additional documentation on the format.

Sintaksė	to_time(string, [format], [language]) [] žymi neprivalomus argumentus
Argumentai	• **string** - string representing a time value • **format** - format used to convert the string into a time • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to convert the string into a time
Pavyzdžiai	• `to_time('12:30:01')` → 12:30:01 • `to_time('12:34','HH:mm')` → 12:34:00 • `to_time('12:34','HH:mm','fr')` → 12:34:00

Further reading: *format_date*

9.2.6 CRS Functions

This group contains functions to operate on coordinate reference system objects.

crs_from_text

Creates a coordinate reference system from a string definition. The string can represent an authority ID, a WKT definition, or a PROJ string definition of the CRS.

Sintaksė	crs_from_text(definition)
Argumentai	• **definition** - CRS definition
Pavyzdžiai	• `crs_from_text('EPSG:3857')` → crs value ‚EPSG:3857 - WGS 84 / Pseudo-Mercator‘ • `crs_from_text('PROJ:+proj=merc +a=6378137 +b=6378137 +lat_ts=0 +lon_0=0 +x_0=0 +y_0=0 +k=1 +units=m +nadgrids=@null +wktext +no_defs +type=crs')` → crs value ‚EPSG:3857 - WGS 84 / Pseudo-Mercator‘

crs_to_authid

Returns the authority ID string for a coordinate reference system.

Sintaksė	crs_to_authid(crs)
Argumentai	• **crs** - crs value
Pavyzdžiai	• `crs_to_authid(crs_from_text('PROJ:+proj=merc +a=6378137 +b=6378137 +lat_ts=0 +lon_0=0 +x_0=0 +y_0=0 +k=1 +units=m +nadgrids=@null +wktext +no_defs +type=crs'))` → ‚EPSG:3857‘

9.2.7 Custom Functions

This group contains functions created by the user. See *Funkcijų redaktorius* for more details.

9.2.8 Date and Time Functions

This group contains functions for handling date and time data. This group shares several functions with the *Conversions Functions* (to_date, to_time, to_datetime, to_interval) and *String Functions* (format_date) groups.

Pastaba

Storing date, datetime and intervals on fields

The ability to store *date*, *time* and *datetime* values directly on fields depends on the data source's provider (e.g., Shapefile accepts *date* format, but not *datetime* or *time* format). The following are some suggestions to overcome this limitation:

- *date*, *datetime* and *time* can be converted and stored in text type fields using the *format_date()* function.
- *Intervals* can be stored in integer or decimal type fields after using one of the date extraction functions (e.g., *day()* to get the interval expressed in days)

age

Returns the difference between two dates or datetimes.

The difference is returned as an `Interval` and needs to be used with one of the following functions in order to extract useful information:

- `year`
- `month`
- `week`
- `day`
- `hour`
- `minute`
- `second`

Sintaksė	age(datetime1, datetime2)
Argumentai	• **datetime1** - a string, date or datetime representing the later date • **datetime2** - a string, date or datetime representing the earlier date
Pavyzdžiai	• `day(age('2012-05-12','2012-05-02'))` → 10 • `hour(age('2012-05-12','2012-05-02'))` → 240

datetime_from_epoch

Returns a datetime whose date and time are the number of milliseconds, msecs, that have passed since 1970-01-01T00:00:00.000, Coordinated Universal Time (Qt.UTC), and converted to Qt.LocalTime.

Sintaksė	datetime_from_epoch(int)
Argumentai	• **int** - number (milliseconds)
Pavyzdžiai	• `datetime_from_epoch(1483225200000)` → 2017-01-01T00:00:00

day

Extracts the day from a date, or the number of days from an interval.

Date variant

Extracts the day from a date or datetime.

Sintaksė	day(date)
Argumentai	• **date** - a date or datetime value
Pavyzdžiai	• `day('2012-05-12')` → 12

Interval variant

Calculates the length in days of an interval.

Sintaksė	day(interval)
Argumentai	• **interval** - interval value to return number of days from
Pavyzdžiai	• `day(to_interval('3 days'))` → 3 • `day(to_interval('3 weeks 2 days'))` → 23 • `day(age('2012-01-01','2010-01-01'))` → 730

day_of_week

Returns the day of the week for a specified date or datetime. The returned value ranges from 0 to 6, where 0 corresponds to a Sunday and 6 to a Saturday.

Sintaksė	day_of_week(date)
Argumentai	• **date** - date or datetime value
Pavyzdžiai	• `day_of_week(to_date('2015-09-21'))` → 1

epoch

Returns the interval in milliseconds between the unix epoch and a given date value.

Sintaksė	epoch(date)
Argumentai	• **date** - a date or datetime value
Pavyzdžiai	• `epoch(to_date('2017-01-01'))` → 1483203600000

format_date

Formats a date type or string into a custom string format. Uses Qt date/time format strings. See QDateTime::toString.

Sintaksė	format_date(datetime, format, [language]) [] žymi neprivalomus argumentus
Argumentai	• **datetime** - date, time or datetime value • **format** - String template used to format the string.

Expression	Output
d	the day as number without a leading zero (1 to 31)
dd	the day as number with a leading zero (01 to 31)
ddd	the abbreviated localized day name (e.g. ‚Mon' to ‚Sun')
dddd	the long localized day name (e.g. ‚Monday' to ‚Sunday')
M	the month as number without a leading zero (1-12)
MM	the month as number with a leading zero (01-12)
MMM	the abbreviated localized month name (e.g. ‚Jan' to ‚Dec')
MMMM	the long localized month name (e.g. ‚January' to ‚December')
yy	the year as two digit number (00-99)
yyyy	the year as four digit number

These expressions may be used for the time part of the format string:

Expression	Output
h	the hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)
hh	the hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)
H	the hour without a leading zero (0 to 23, even with AM/PM display)
HH	the hour with a leading zero (00 to 23, even with AM/PM display)
m	the minute without a leading zero (0 to 59)
mm	the minute with a leading zero (00 to 59)
s	the second without a leading zero (0 to 59)
ss	the second with a leading zero (00 to 59)
z	the milliseconds without trailing zeroes (0 to 999)
zzz	the milliseconds with trailing zeroes (000 to 999)
AP or A	interpret as an AM/PM time. *AP* must be either ‚AM' or ‚PM'.
ap or a	Interpret as an AM/PM time. *ap* must be either ‚am' or ‚pm'.

- **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to format the date into a custom string. By default the current QGIS user locale is used.

Pavyzdžiai

- `format_date('2012-05-15','dd.MM.yyyy')` → ‚15.05.2012'
- `format_date('2012-05-15','d MMMM yyyy','fr')` → ‚15 mai 2012'
- `format_date('2012-05-15','dddd')` → ‚Tuesday', if the current locale is an English variant
- `format_date('2012-05-15 13:54:20','dd.MM.yy')` → ‚15.05.12'
- `format_date('13:54:20','hh:mm AP')` → ‚01:54 PM'

hour

Extracts the hour part from a datetime or time, or the number of hours from an interval.

Time variant

Extracts the hour part from a time or datetime.

Sintaksė	hour(datetime)
Argumentai	• **datetime** - a time or datetime value
Pavyzdžiai	• `hour( to_datetime('2012-07-22 13:24:57') )` → 13

Interval variant

Calculates the length in hours of an interval.

Sintaksė	hour(interval)
Argumentai	• **interval** - interval value to return number of hours from
Pavyzdžiai	• `hour(to_interval('3 hours'))` → 3 • `hour(age('2012-07-22T13:00:00','2012-07-22T10:00:00'))` → 3 • `hour(age('2012-01-01','2010-01-01'))` → 17520

make_date

Creates a date value from year, month and day numbers.

Sintaksė	make_date(year, month, day)
Argumentai	• **year** - Year number. Years 1 to 99 are interpreted as is. Year 0 is invalid. • **month** - Month number, where 1=January • **day** - Day number, beginning with 1 for the first day in the month
Pavyzdžiai	• `make_date(2020,5,4)` → date value 2020-05-04

make_datetime

Creates a datetime value from year, month, day, hour, minute and second numbers.

Sintaksė	make_datetime(year, month, day, hour, minute, second)
Argumentai	• **year** - Year number. Years 1 to 99 are interpreted as is. Year 0 is invalid. • **month** - Month number, where 1=January • **day** - Day number, beginning with 1 for the first day in the month • **hour** - Hour number • **minute** - Minutes • **second** - Seconds (fractional values include milliseconds)
Pavyzdžiai	• `make_datetime(2020,5,4,13,45,30.5)` → datetime value 2020-05-04 13:45:30.500

make_interval

Creates an interval value from year, month, weeks, days, hours, minute and seconds values.

Sintaksė	make_interval([years=0], [months=0], [weeks=0], [days=0], [hours=0], [minutes=0], [seconds=0]) [] žymi neprivalomus argumentus
Argumentai	• **years** - Number of years (assumes a 365.25 day year length). • **months** - Number of months (assumes a 30 day month length) • **weeks** - Number of weeks • **days** - Number of days • **hours** - Number of hours • **minutes** - Number of minutes • **seconds** - Number of seconds
Pavyzdžiai	• `make_interval(hours:=3)` → interval: 3 hours • `make_interval(days:=2, hours:=3)` → interval: 2.125 days • `make_interval(minutes:=0.5, seconds:=5)` → interval: 35 seconds

make_time

Creates a time value from hour, minute and second numbers.

Sintaksė	make_time(hour, minute, second)
Argumentai	• **hour** - Hour number • **minute** - Minutes • **second** - Seconds (fractional values include milliseconds)
Pavyzdžiai	• `make_time(13,45,30.5)` → time value 13:45:30.500

minute

Extracts the minutes part from a datetime or time, or the number of minutes from an interval.

Time variant

Extracts the minutes part from a time or datetime.

Sintaksė	minute(datetime)
Argumentai	• **datetime** - a time or datetime value
Pavyzdžiai	• `minute( to_datetime('2012-07-22 13:24:57') )` → 24

Interval variant

Calculates the length in minutes of an interval.

Sintaksė	minute(interval)
Argumentai	• **interval** - interval value to return number of minutes from
Pavyzdžiai	• `minute(to_interval('3 minutes'))` → 3 • `minute(age('2012-07-22T00:20:00','2012-07-22T00:00:00'))` → 20 • `minute(age('2012-01-01','2010-01-01'))` → 1051200

month

Extracts the month part from a date, or the number of months from an interval.

Date variant

Extracts the month part from a date or datetime.

Sintaksė	month(date)
Argumentai	• **date** - a date or datetime value
Pavyzdžiai	• `month('2012-05-12')` → 05

Interval variant

Calculates the length in months of an interval.

Sintaksė	month(interval)
Argumentai	• **interval** - interval value to return number of months from
Pavyzdžiai	• `month(to_interval('3 months'))` → 3 • `month(age('2012-01-01','2010-01-01'))` → 4.03333

now

Returns the current date and time. The function is static and will return consistent results while evaluating. The time returned is the time when the expression is prepared.

Sintaksė	now()
Pavyzdžiai	• `now()` → 2012-07-22T13:24:57

second

Extracts the seconds part from a datetime or time, or the number of seconds from an interval.

Time variant

Extracts the seconds part from a time or datetime.

Sintaksė	second(datetime)
Argumentai	• **datetime** - a time or datetime value
Pavyzdžiai	• `second( to_datetime('2012-07-22 13:24:57') )` → 57

Interval variant

Calculates the length in seconds of an interval.

Sintaksė	second(interval)
Argumentai	• **interval** - interval value to return number of seconds from
Pavyzdžiai	• `second(to_interval('3 minutes'))` → 180 • `second(age('2012-07-22T00:20:00','2012-07-22T00:00:00'))` → 1200 • `second(age('2012-01-01','2010-01-01'))` → 63072000

to_date

Converts a string into a date object. An optional format string can be provided to parse the string; see QDate::fromString or the documentation of the format_date function for additional documentation on the format. By default the current QGIS user locale is used.

Sintaksė	to_date(string, [format], [language]) [] žymi neprivalomus argumentus
Argumentai	• **string** - string representing a date value • **format** - format used to convert the string into a date • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to convert the string into a date. By default the current QGIS user locale is used.
Pavyzdžiai	• `to_date('2012-05-04')` → 2012-05-04 • `to_date('June 29, 2019','MMMM d, yyyy')` → 2019-06-29, if the current locale uses the name ‚June' for the sixth month, otherwise an error occurs • `to_date('29 juin, 2019','d MMMM, yyyy','fr')` → 2019-06-29

Further reading: *format_date*

to_datetime

Converts a string into a datetime object. An optional format string can be provided to parse the string; see QDate::fromString, QTime::fromString or the documentation of the format_date function for additional documentation on the format. By default the current QGIS user locale is used.

Sintaksė	to_datetime(string, [format], [language]) [] žymi neprivalomus argumentus
Argumentai	• **string** - string representing a datetime value • **format** - format used to convert the string into a datetime • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to convert the string into a datetime. By default the current QGIS user locale is used.
Pavyzdžiai	• `to_datetime('2012-05-04 12:50:00')` → 2012-05-04T12:50:00 • `to_datetime('June 29, 2019 @ 12:34','MMMM d, yyyy @ HH:mm')` → 2019-06-29T12:34, if the current locale uses the name ‚June' for the sixth month, otherwise an error occurs • `to_datetime('29 juin, 2019 @ 12:34','d MMMM, yyyy @ HH:mm', 'fr')` → 2019-06-29T12:34

Further reading: *format_date*

to_interval

Converts a string to an interval type. Can be used to take days, hours, month, etc of a date.

Sintaksė	to_interval(string)
Argumentai	• **string** - a string representing an interval. Allowable formats include {n} days {n} hours {n} months.
Pavyzdžiai	• `to_interval('1 day 2 hours')` → interval: 1.08333 days • `to_interval( '0.5 hours' )` → interval: 30 minutes • `to_datetime('2012-05-05 12:00:00') - to_interval('1 day 2 hours')` → 2012-05-04T10:00:00

to_time

Converts a string into a time object. An optional format string can be provided to parse the string; see QTime::fromString for additional documentation on the format.

Sintaksė	to_time(string, [format], [language]) [] žymi neprivalomus argumentus
Argumentai	• **string** - string representing a time value • **format** - format used to convert the string into a time • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to convert the string into a time
Pavyzdžiai	• `to_time('12:30:01')` → 12:30:01 • `to_time('12:34','HH:mm')` → 12:34:00 • `to_time('12:34','HH:mm','fr')` → 12:34:00

Further reading: *format_date*

week

Extracts the week number from a date, or the number of weeks from an interval.

Date variant

Extracts the week number from a date or datetime.

Sintaksė	week(date)
Argumentai	• **date** - a date or datetime value
Pavyzdžiai	• `week('2012-05-12')` → 19

Interval variant

Calculates the length in weeks of an interval.

Sintaksė	week(interval)
Argumentai	• **interval** - interval value to return number of months from
Pavyzdžiai	• `week(to_interval('3 weeks'))` → 3 • `week(age('2012-01-01','2010-01-01'))` → 104.285

year

Extracts the year part from a date, or the number of years from an interval.

Date variant

Extracts the year part from a date or datetime.

Sintaksė	year(date)
Argumentai	• **date** - a date or datetime value
Pavyzdžiai	• `year('2012-05-12')` → 2012

Interval variant

Calculates the length in years of an interval.

Sintaksė	year(interval)
Argumentai	• **interval** - interval value to return number of years from
Pavyzdžiai	• `year(to_interval('3 years'))` → 3 • `year(age('2012-01-01','2010-01-01'))` → 1.9986

Some examples:

Besides these functions, subtracting dates, datetimes or times using the − (minus) operator will return an interval.

Adding or subtracting an interval to dates, datetimes or times, using the + (plus) and − (minus) operators, will return a datetime.

- Get the number of days until QGIS 3.0 release:

```
to_date('2017-09-29') - to_date(now())
-- Returns <interval: 203 days>
```

- The same with time:

```
to_datetime('2017-09-29 12:00:00') - now()
-- Returns <interval: 202.49 days>
```

- Get the datetime of 100 days from now:

```
now() + to_interval('100 days')
-- Returns <datetime: 2017-06-18 01:00:00>
```

9.2.9 Fields and Values

Contains a list of fields from the active layer, and special values. Fields list includes the ones stored in the dataset, *virtual* and *auxiliary* ones as well as from *joins*.

Double-click a field name to have it added to your expression. You can also type the field name (preferably inside double quotes) or its *alias*.

To retrieve fields values to use in an expression, select the appropriate field and, in the shown widget, choose between *10 Samples* and *All Unique*. Requested values are then displayed and you can use the *Search* box at the top of the list to filter the result. Sample values can also be accessed via right-clicking on a field.

To add a value to the expression you are writing, double-click on it in the list. If the value is of a string type, it should be simple quoted, otherwise no quote is needed.

NULL

Equates to a NULL value.

Sintaksė	NULL
Pavyzdžiai	• `NULL` → a NULL value

Pastaba

To test for NULL use an *IS NULL* or *IS NOT NULL* expression.

9.2.10 Files and Paths Functions

This group contains functions which manipulate file and path names.

base_file_name

Returns the base name of the file without the directory or file suffix.

Sintaksė	base_file_name(path)
Argumentai	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Pavyzdžiai	• `base_file_name('/home/qgis/data/country_boundaries.shp')` → ‚country_boundaries‘

exif

Retrieves exif tag values from an image file.

Sintaksė	exif(path, [tag]) [] žymi neprivalomus argumentus
Argumentai	• **path** - An image file path or a map layer value. If a map layer value is specified then the file source of the layer will be used. • **tag** - The tag to return. If empty, a map with all exif tag values will be returned.
Pavyzdžiai	• `exif('/my/photo.jpg','Exif.Image.Orientation')` → 0

file_exists

Returns TRUE if a file path exists.

Sintaksė	file_exists(path)
Argumentai	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Pavyzdžiai	• `file_exists('/home/qgis/data/country_boundaries.shp')` → TRUE

file_name

Returns the name of a file (including the file extension), excluding the directory.

Sintaksė	file_name(path)
Argumentai	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Pavyzdžiai	• `file_name('/home/qgis/data/country_boundaries.shp')` → ‚country_boundaries.shp‘

file_path

Returns the directory component of a file path. This does not include the file name.

Sintaksė	file_path(path)
Argumentai	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Pavyzdžiai	• `file_path('/home/qgis/data/country_boundaries.shp')` → ‚/home/qgis/data‘

file_size

Returns the size (in bytes) of a file.

Sintaksė	file_size(path)
Argumentai	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Pavyzdžiai	• `file_size('/home/qgis/data/country_boundaries.geojson')` → 5674

file_suffix

Returns the file suffix (extension) from a file path.

Sintaksė	file_suffix(path)
Argumentai	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Pavyzdžiai	• `file_suffix('/home/qgis/data/country_boundaries.shp')` →‚shp‘

is_directory

Returns TRUE if a path corresponds to a directory.

Sintaksė	is_directory(path)
Argumentai	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Pavyzdžiai	• `is_directory('/home/qgis/data/country_boundaries.shp')` → FALSE • `is_directory('/home/qgis/data/')` → TRUE

is_file

Returns TRUE if a path corresponds to a file.

Sintaksė	is_file(path)
Argumentai	• **path** - a file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Pavyzdžiai	• `is_file('/home/qgis/data/country_boundaries.shp')` → TRUE • `is_file('/home/qgis/data/')` → FALSE

9.2.11 Form Functions

This group contains functions that operate exclusively under the attribute form context. For example, in *field's widgets* settings.

current_parent_value

Only usable in an embedded form context, this function returns the current, unsaved value of a field in the parent form currently being edited. This will differ from the parent feature's actual attribute values for features which are currently being edited or have not yet been added to a parent layer. When used in a value-relation widget filter expression, this function should be wrapped into a ‚coalesce()' that can retrieve the actual parent feature from the layer when the form is not used in an embedded context.

Sintaksė	current_parent_value(field_name)
Argumentai	• **field_name** - a field name in the current parent form
Pavyzdžiai	• `current_parent_value( 'FIELD_NAME' )` → The current value of a field ‚FIELD_NAME' in the parent form.

current_value

Returns the current, unsaved value of a field in the form or table row currently being edited. This will differ from the feature's actual attribute values for features which are currently being edited or have not yet been added to a layer.

Sintaksė	current_value(field_name)
Argumentai	• **field_name** - a field name in the current form or table row
Pavyzdžiai	• `current_value( 'FIELD_NAME' )` → The current value of field ‚FIELD_NAME'.

9.2.12 Fuzzy Matching Functions

This group contains functions for fuzzy comparisons between values.

hamming_distance

Returns the Hamming distance between two strings. This equates to the number of characters at corresponding positions within the input strings where the characters are different. The input strings must be the same length, and the comparison is case-sensitive.

Sintaksė	hamming_distance(string1, string2)
Argumentai	• **string1** - a string • **string2** - a string
Pavyzdžiai	• `hamming_distance('abc','xec')` → 2 • `hamming_distance('abc','ABc')` → 2 • `hamming_distance(upper('abc'),upper('ABC'))` → 0 • `hamming_distance('abc','abcd')` → NULL

levenshtein

Returns the Levenshtein edit distance between two strings. This equates to the minimum number of character edits (insertions, deletions or substitutions) required to change one string to another.

The Levenshtein distance is a measure of the similarity between two strings. Smaller distances mean the strings are more similar, and larger distances indicate more different strings. The distance is case sensitive.

Sintaksė	levenshtein(string1, string2)
Argumentai	• **string1** - a string • **string2** - a string
Pavyzdžiai	• `levenshtein('kittens','mitten')` → 2 • `levenshtein('Kitten','kitten')` → 1 • `levenshtein(upper('Kitten'),upper('kitten'))` → 0

longest_common_substring

Returns the longest common substring between two strings. This substring is the longest string that is a substring of the two input strings. For example, the longest common substring of „ABABC“ and „BABCA“ is „BABC“. The substring is case sensitive.

Sintaksė	longest_common_substring(string1, string2)
Argumentai	• **string1** - a string • **string2** - a string
Pavyzdžiai	• `longest_common_substring('ABABC','BABCA')` → ‚BABC‘ • `longest_common_substring('abcDeF','abcdef')` → ‚abc‘ • `longest_common_substring(upper('abcDeF'),upper('abcdex'))` → ‚ABCDE‘

soundex

Returns the Soundex representation of a string. Soundex is a phonetic matching algorithm, so strings with similar sounds should be represented by the same Soundex code.

Sintaksė	soundex(string)
Argumentai	• **string** - a string
Pavyzdžiai	• `soundex('robert')` → ‚R163‘ • `soundex('rupert')` → ‚R163‘ • `soundex('rubin')` → ‚R150‘

9.2.13 General Functions

This group contains general assorted functions.

env

Gets an environment variable and returns its content as a string. If the variable is not found, NULL will be returned. This is handy to inject system specific configuration like drive letters or path prefixes. Definition of environment variables depends on the operating system, please check with your system administrator or the operating system documentation how this can be set.

Sintaksė	env(name)
Argumentai	• **name** - The name of the environment variable which should be retrieved.
Pavyzdžiai	• `env( 'LANG' )` → ‚en_US.UTF-8‘ • `env( 'MY_OWN_PREFIX_VAR' )` → ‚Z:‘ • `env( 'I_DO_NOT_EXIST' )` → NULL

eval

Evaluates an expression which is passed in a string. Useful to expand dynamic parameters passed as context variables or fields.

Sintaksė	eval(expression)
Argumentai	• **expression** - an expression string
Pavyzdžiai	• `eval('\'nice\'')` → ‚nice‘ • `eval(@expression_var)` → [whatever the result of evaluating @expression_var might be…]

eval_template

Evaluates a template which is passed in a string. Useful to expand dynamic parameters passed as context variables or fields.

Sintaksė	eval_template(template)
Argumentai	• **template** - a template string
Pavyzdžiai	• `eval_template('QGIS [% upper(\'rocks\') %]')` → QGIS ROCKS

is_layer_visible

Returns TRUE if a specified layer is visible.

Sintaksė	is_layer_visible(layer)
Argumentai	• **layer** - tekstas, reprezentuojantis arba sluoksnio pavadinimą, arba sluoksnio ID
Pavyzdžiai	• `is_layer_visible('baseraster')` → TRUE

mime_type

Returns the mime type of the binary data.

Sintaksė	mime_type(bytes)
Argumentai	• **bytes** - the binary data
Pavyzdžiai	• `mime_type('<html><body></body></html>')` → text/html • `mime_type(from_base64('R0lGODlhAQABAAAAACH5BAEKAAEALAAAAAABAAEAAAIAOw==` → image/gif

var

Returns the value stored within a specified variable.

Sintaksė	var(name)
Argumentai	• **name** - a variable name
Pavyzdžiai	• `var('qgis_version')` → ‚2.12‘

Further reading: List of default *variables*

with_variable

This function sets a variable for any expression code that will be provided as 3rd argument. This is only useful for complicated expressions, where the same calculated value needs to be used in different places.

Sintaksė	with_variable(name, value, expression)
Argumentai	• **name** - the name of the variable to set • **value** - the value to set • **expression** - the expression for which the variable will be available
Pavyzdžiai	• `with_variable('my_sum', 1 + 2 + 3, @my_sum * 2 + @my_sum * 5)` → 42

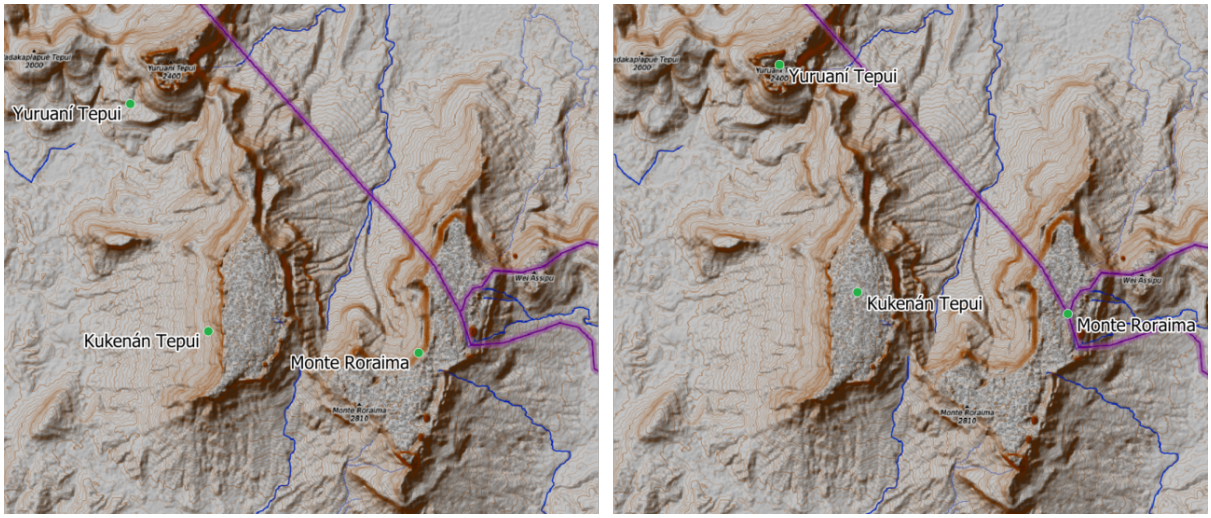
9.2.14 Geometry Functions

This group contains functions that operate on geometry objects (e.g. buffer, transform, $area).

affine_transform

Returns the geometry after an affine transformation. Calculations are in the Spatial Reference System of this geometry. The operations are performed in a scale, rotation, translation order. If there is a Z or M offset but the coordinate is not present in the geometry, it will be added.

Sintaksė	affine_transform(geometry, delta_x, delta_y, rotation_z, scale_x, scale_y, [delta_z=0], [delta_m=0], [scale_z=1], [scale_m=1]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **delta_x** - x-axis translation • **delta_y** - y-axis translation • **rotation_z** - rotation around z-axis in degrees counter-clockwise • **scale_x** - x-axis scale factor • **scale_y** - y-axis scale factor • **delta_z** - z-axis translation • **delta_m** - m-axis translation • **scale_z** - z-axis scale factor • **scale_m** - m-axis scale factor
Pavyzdžiai	• `geom_to_wkt(affine_transform(geom_from_wkt('LINESTRING(1 1, 2 2)'), 2, 2, 0, 1, 1))` → ‚LineString (3 3, 4 4)‘ • `geom_to_wkt(affine_transform(geom_from_wkt('POLYGON((0 0, 0 3, 2 2, 0 0))'), 0, 0, -90, 1, 2))` → ‚Polygon ((0 0, 6 0, 4 -2, 0 0))‘ • `geom_to_wkt(affine_transform(geom_from_wkt('POINT(3 1)'), 0, 0, 0, 1, 1, 5, 0))` → ‚PointZ (3 1 5)‘



9.4 Fig. : Vektorinis taškų sluoksnis (žali taškai) prieš (kairėje) ir po (dešinėje) paprastos transformacijos (vertimo).

angle_at_vertex

Returns the bisector angle (average angle) to the geometry for a specified vertex on a linestring geometry. Angles are in degrees clockwise from north.

Sintaksė	angle_at_vertex(geometry, vertex)
Argumentai	• **geometry** - a linestring geometry • **vertex** - vertex index, starting from 0; if the value is negative, the selected vertex index will be its total count minus the absolute value
Pavyzdžiai	• `angle_at_vertex(geometry:=geom_from_wkt('LineString(0 0, 10 0, 10 10)'),vertex:=1)` → 45.0

apply_dash_pattern

Applies a dash pattern to a geometry, returning a MultiLineString geometry which is the input geometry stroked along each line/ring with the specified pattern.

Sintaksė	apply_dash_pattern(geometry, pattern, [start_rule=no_rule], [end_rule=no_rule], [adjustment=both], [pattern_offset=0]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry (accepts (multi)linestrings or (multi)polygons). • **pattern** - dash pattern, as an array of numbers representing dash and gap lengths. Must contain an even number of elements. • **start_rule** - optional rule for constraining the start of the pattern. Valid values are ‚no_rule', ‚full_dash', ‚half_dash', ‚full_gap', ‚half_gap'. • **end_rule** - optional rule for constraining the end of the pattern. Valid values are ‚no_rule', ‚full_dash', ‚half_dash', ‚full_gap', ‚half_gap'. • **adjustment** - optional rule for specifying which part of patterns are adjusted to fit the desired pattern rules. Valid values are ‚both', ‚dash', ‚gap'. • **pattern_offset** - Optional distance specifying a specific distance along the pattern to commence at.
Pavyzdžiai	• `geom_to_wkt(apply_dash_pattern(geom_from_wkt('LINESTRING(1 1, 10 1)'), array(3, 1)))` → MultiLineString ((1 1, 4 1),(5 1, 8 1),(9 1, 10 1, 10 1)) • `geom_to_wkt(apply_dash_pattern(geom_from_wkt('LINESTRING(1 1, 10 1)'), array(3, 1), start_rule:='half_dash'))` → MultiLineString ((1 1, 2.5 1),(3.5 1, 6.5 1),(7.5 1, 10 1, 10 1))

$area

Returns the area of the current feature. The area calculated by this function respects both the current project's ellipsoid setting and area unit settings. For example, if an ellipsoid has been set for the project then the calculated area will be ellipsoidal, and if no ellipsoid is set then the calculated area will be planimetric.

Sintaksė	$area
Pavyzdžiai	• `$area` → 42

area

Returns the area of a geometry polygon object. Calculations are always planimetric in the Spatial Reference System (SRS) of this geometry, and the units of the returned area will match the units for the SRS. This differs from the calculations performed by the $area function, which will perform ellipsoidal calculations based on the project's ellipsoid and area unit settings.

Sintaksė	area(geometry)
Argumentai	• **geometry** - polygon geometry object
Pavyzdžiai	• `area(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'))` → 8.0

azimuth

Returns the north-based azimuth as the angle in radians measured clockwise from the vertical on point_a to point_b.

Sintaksė	azimuth(point_a, point_b)
Argumentai	• **point_a** - point geometry • **point_b** - point geometry
Pavyzdžiai	• `degrees( azimuth( make_point(25, 45), make_point(75, 100) ) )` → 42.273689 • `degrees( azimuth( make_point(75, 100), make_point(25,45) ) )` → 222.273689

bearing

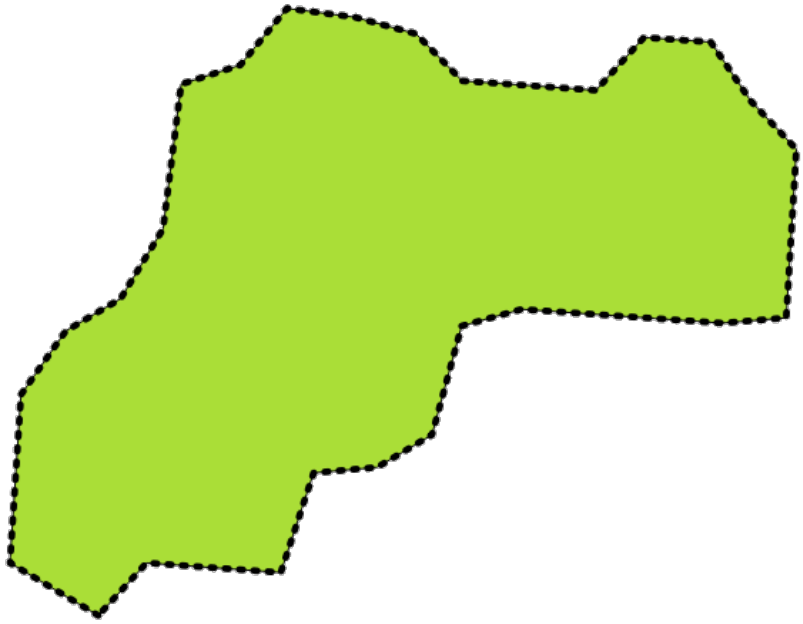
Returns the north-based bearing as the angle in radians measured clockwise on the ellipsoid from the vertical on point_a to point_b.

Sintaksė	bearing(point_a, point_b, [source_crs], [ellipsoid]) [] žymi neprivalomus argumentus
Argumentai	• **point_a** - point geometry • **point_b** - point geometry • **source_crs** - an optional string or CRS object representing the source CRS of the points. By default the current layer's CRS is used. • **ellipsoid** - an optional string representing the acronym or the authority ID (e.g., ‚EPSG:7030') of the ellipsoid on which the bearing should be measured. By default the current project's ellipsoid setting is used.
Pavyzdžiai	• `degrees( bearing( make_point(16198544, -4534850), make_point(18736872, -1877769), 'EPSG:3857', 'EPSG:7030') )` → 49.980071 • `degrees( bearing( make_point(18736872, -1877769), make_point(16198544, -4534850), 'EPSG:3857', 'WGS84') )` → 219.282386

boundary

Returns the closure of the combinatorial boundary of the geometry (ie the topological boundary of the geometry). For instance, a polygon geometry will have a boundary consisting of the linestrings for each ring in the polygon. Some geometry types do not have a defined boundary, e.g., points or geometry collections, and will return NULL.

Sintaksė	boundary(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `geom_to_wkt(boundary(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')))` → ‚LineString(1 1,0 0,-1 1,1 1)' • `geom_to_wkt(boundary(geom_from_wkt('LineString(1 1,0 0,-1 1)')))` → ‚MultiPoint ((1 1),(-1 1))'



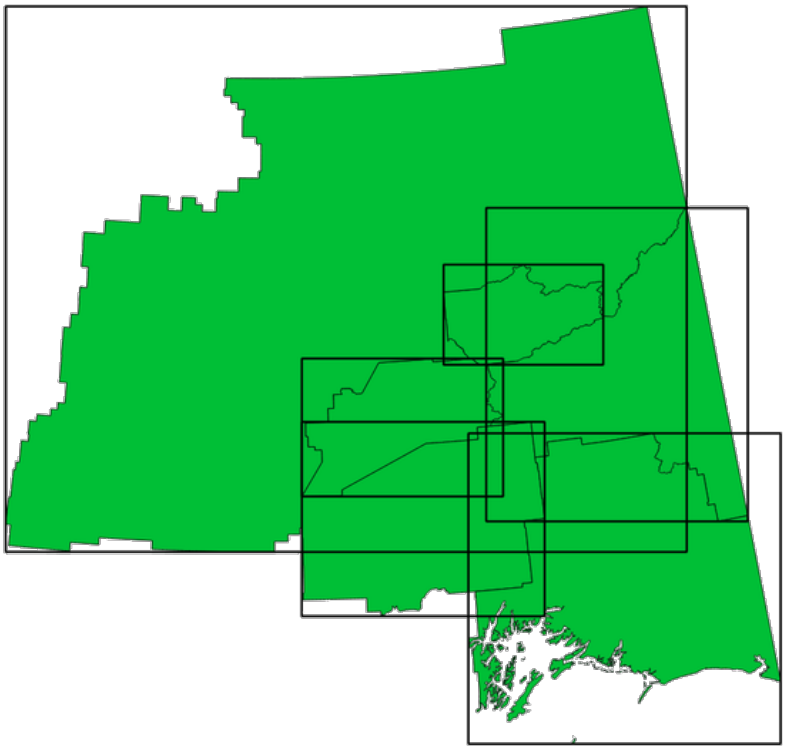
9.5 Fig. : Boundary (black dashed line) of the source polygon layer

Further reading: *Ribos* algorithm

bounds

Returns a geometry which represents the bounding box of an input geometry. Calculations are in the Spatial Reference System of this geometry.

Sintaksė	bounds(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `bounds(@geometry)` → bounding box of the current feature's geometry • `geom_to_wkt(bounds(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')))` → ‚Polygon ((-1 0, 1 0, 1 1, -1 1, -1 0))‘



9.6 Fig. : Juodos linijos vaizduoja kiekvieno poligono geoobjekto apimties stačiakampį

Further reading: *Apimties stačiakampiai* algorithm

bounds_height

Returns the height of the bounding box of a geometry. Calculations are in the Spatial Reference System of this geometry.

Sintaksė	bounds_height(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `bounds_height(@geometry)` → height of bounding box of the current feature's geometry • `bounds_height(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))'))` → 1

bounds_width

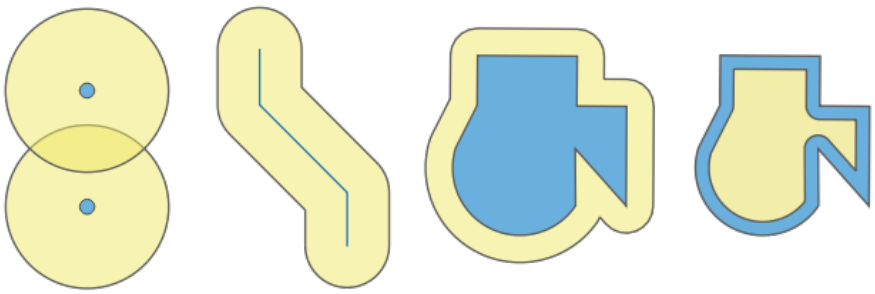
Returns the width of the bounding box of a geometry. Calculations are in the Spatial Reference System of this geometry.

Sintaksė	bounds_width(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `bounds_width(@geometry)` → width of bounding box of the current feature's geometry • `bounds_width(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))'))` → 2

buffer

Returns a geometry that represents all points whose distance from this geometry is less than or equal to distance. Calculations are in the Spatial Reference System of this geometry.

Sintaksė	buffer(geometry, distance, [segments=8], [cap='round'], [join='round'], [miter_limit=2]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **distance** - buffer distance in layer units • **segments** - number of segments to use to represent a quarter circle when a round join style is used. A larger number results in a smoother buffer with more nodes. • **cap** - end cap style for buffer. Valid values are ‚round', ‚flat' or ‚square' • **join** - join style for buffer. Valid values are ‚round', ‚bevel' or ‚miter'. • **miter_limit** - miter distance limit, for use when the join style is set to ‚miter'
Pavyzdžiai	• `buffer(@geometry, 10.5)` → polygon of the current feature's geometry buffered by 10.5 units



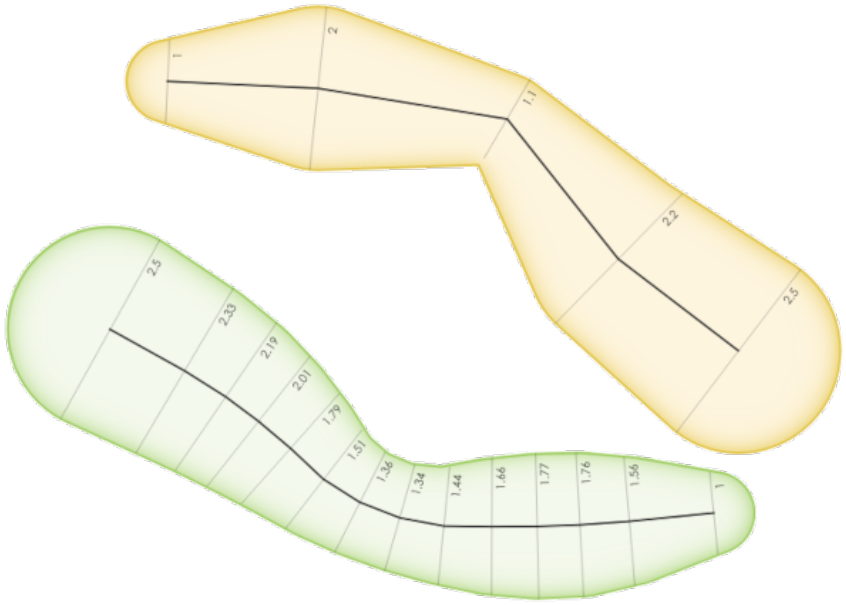
9.7 Fig. : Buferiai (geltoni) taškų, linijų, poligonų su teigiamu buferiu ir poligonas su neigiamu buferiu

Further reading: *Buferis* algorithm

buffer_by_m

Creates a buffer along a line geometry where the buffer diameter varies according to the m-values at the line vertices.

Sintaksė	buffer_by_m(geometry, [segments=8]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - input geometry. Must be a (multi)line geometry with m values. • **segments** - number of segments to approximate quarter-circle curves in the buffer.
Pavyzdžiai	• `buffer_by_m(geometry:=geom_from_wkt('LINESTRINGM(1 2 0.5, 4 2 0.2)'),segments:=8)` → A variable width buffer starting with a diameter of 0.5 and ending with a diameter of 0.2 along the linestring geometry.



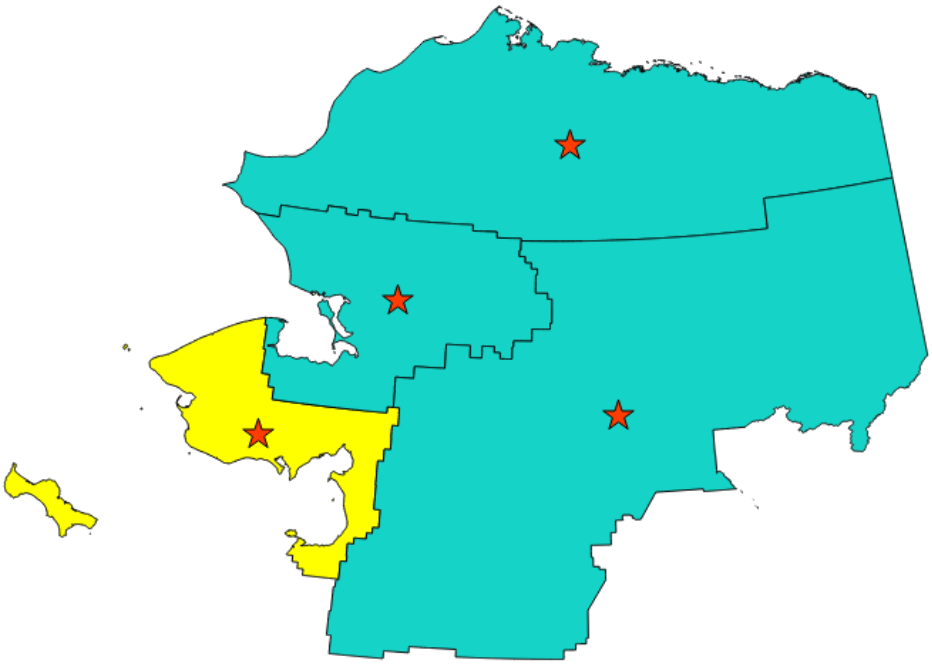
9.8 Fig. : Buffering line features using the m value on the vertices

Further reading: *Variable width buffer (by M value)* algorithm

centroid

Returns the geometric center of a geometry.

Sintaksė	centroid(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `centroid(@geometry)` → a point geometry



9.9 Fig. : Raudonos žvaigždės reprezentuoja įvesties sluoksnio geoobjektų centroidus.

Further reading: *Centroidai* algorithm

close_line

Returns a closed line string of the input line string by appending the first point to the end of the line, if it is not already closed. If the geometry is not a line string or multi line string then the result will be NULL.

Sintaksė	close_line(geometry)
Argumentai	• **geometry** - a line string geometry
Pavyzdžiai	• `geom_to_wkt(close_line(geom_from_wkt('LINESTRING(0 0, 1 0, 1 1)')))` → ‚LineString (0 0, 1 0, 1 1, 0 0)‘ • `geom_to_wkt(close_line(geom_from_wkt('LINESTRING(0 0, 1 0, 1 1, 0 0)')))` → ‚LineString (0 0, 1 0, 1 1, 0 0)‘

closest_point

Returns the point on geometry1 that is closest to geometry2.

Sintaksė	closest_point(geometry1, geometry2)
Argumentai	• **geometry1** - geometry to find closest point on • **geometry2** - geometry to find closest point to
Pavyzdžiai	• `geom_to_wkt(closest_point(geom_from_wkt('LINESTRING (20 80, 98 190, 110 180, 50 75 )'),geom_from_wkt('POINT(100 100)')))` → ‚Point(73.0769 115.384)‘

collect_geometries

Collects a set of geometries into a multi-part geometry object.

List of arguments variant

Geometry parts are specified as separate arguments to the function.

Sintaksė	collect_geometries(geometry1, geometry2, …)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `geom_to_wkt(collect_geometries(make_point(1,2), make_point(3,4), make_point(5,6)))` → ‚MultiPoint ((1 2),(3 4),(5 6))‘

Array variant

Geometry parts are specified as an array of geometry parts.

Sintaksė	collect_geometries(array)
Argumentai	• **array** - array of geometry objects
Pavyzdžiai	• `geom_to_wkt(collect_geometries(array(make_point(1,2), make_point(3,4), make_point(5,6)))` → ‚MultiPoint ((1 2),(3 4),(5 6))‘

Further reading: *Collect geometries* algorithm

combine

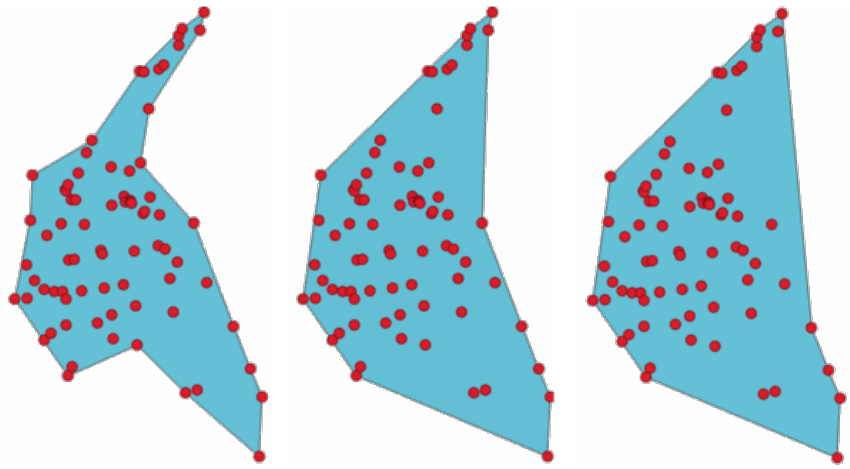
Returns the combination of two geometries.

Sintaksė	combine(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `geom_to_wkt( combine( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 2 1)' ) ) )` → ‚MULTILINESTRING((4 4, 2 1), (3 3, 4 4), (4 4, 5 5))‘ • `geom_to_wkt( combine( geom_from_wkt( 'LINESTRING(3 3, 4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 6 6, 2 1)' ) ) )` → ‚LINESTRING(3 3, 4 4, 6 6, 2 1)‘

concave_hull

Returns a possibly concave polygon that contains all the points in the geometry

Sintaksė	concave_hull(geometry, target_percent, [allow_holes=False]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **target_percent** - the percentage of area of the convex hull the solution tries to approach. A target_percent of 1 gives the same result as the convex hull. A target_percent between 0 and 0.99 produces a result that should have a smaller area than the convex hull. • **allow_holes** - optional argument specifying whether to allow holes within the output geometry. Defaults to FALSE, set to TRUE to allow including holes in the output geometry.
Pavyzdžiai	• `geom_to_wkt(concave_hull(geom_from_wkt('MULTILINESTRING((106 164,30 112,74 70,82 112,130 94,130 62,122 40,156 32,162 76,172 88),(132 178,134 148,128 136,96 128,132 108,150 130,170 142,174 110,156 96,158 90,158 88),(22 64,66 28,94 38,94 68,114 76,112 30,132 10,168 18,178 34,186 52,184 74,190 100,190 122,182 148,178 170,176 184,156 164,146 178,132 186,92 182,56 158,36 150,62 150,76 128,88 118))'), 0.99))` → ‚Polygon ((30 112, 36 150, 92 182, 132 186, 176 184, 190 122, 190 100, 186 52, 178 34, 168 18, 132 10, 112 30, 66 28, 22 64, 30 112))‘



9.10 Fig. : Concave hulls with increasing target_percent parameter

Further reading: *convex_hull*, *Concave hull* algorithm

contains

Tests whether a geometry contains another. Returns TRUE if and only if no points of geometry2 lie in the exterior of geometry1, and at least one point of the interior of geometry2 lies in the interior of geometry1.

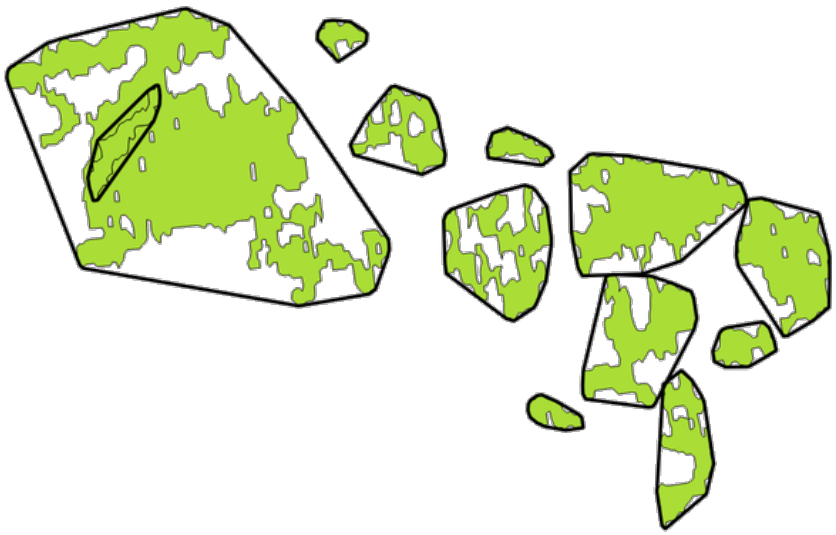
Sintaksė	contains(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `contains( geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))' ), geom_from_wkt( 'POINT(0.5 0.5 )' ) )` → TRUE • `contains( geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → FALSE

Further reading: *overlay_contains*

convex_hull

Returns the convex hull of a geometry. It represents the minimum convex geometry that encloses all geometries within the set.

Sintaksė	convex_hull(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `geom_to_wkt( convex_hull( geom_from_wkt( 'LINESTRING(3 3, 4 4, 4 10)' ) ) )` → ‚POLYGON((3 3, 4 10, 4 4, 3 3))‘



9.11 Fig. : Black lines identify the convex hull for each feature

Further reading: *concave_hull*, *Convex hull* algorithm

crosses

Tests whether a geometry crosses another. Returns TRUE if the supplied geometries have some, but not all, interior points in common.

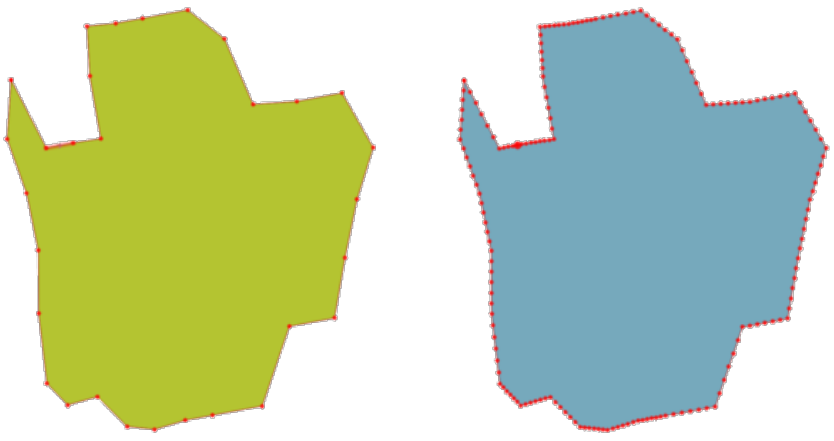
Sintaksė	crosses(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `crosses( geom_from_wkt( 'LINESTRING(3 5, 4 4, 5 3)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `crosses( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → FALSE

Further reading: *overlay_crosses*

densify_by_count

Takes a polygon or line layer geometry and generates a new one in which the geometries have a larger number of vertices than the original one.

Sintaksė	densify_by_count(geometry, vertices)
Argumentai	• **geometry** - a geometry (accepts (multi)linestrings or (multi)polygons). • **vertices** - number of vertices to add (per segment)
Pavyzdžiai	• `geom_to_wkt(densify_by_count(geom_from_wkt('LINESTRING(1 1, 10 1)'), 3))` → LineString (1 1, 3.25 1, 5.5 1, 7.75 1, 10 1)



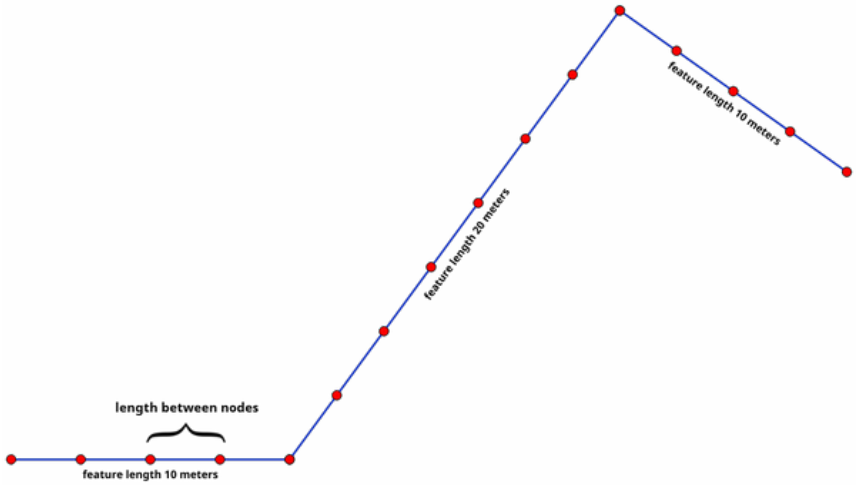
9.12 Fig. : Red points show the vertices before and after the densify

Further reading: *Densify by count* algorithm

densify_by_distance

Takes a polygon or line layer geometry and generates a new one in which the geometries are densified by adding additional vertices on edges that have a maximum distance of the specified interval distance.

Sintaksė	densify_by_distance(geometry, distance)
Argumentai	• **geometry** - a geometry (accepts (multi)linestrings or (multi)polygons). • **distance** - maximum interval distance between vertices in output geometry
Pavyzdžiai	• `geom_to_wkt(densify_by_distance(geom_from_wkt('LINESTRING(1 1, 10 1)'), 4))` → LineString (1 1, 4 1, 7 1, 10 1)



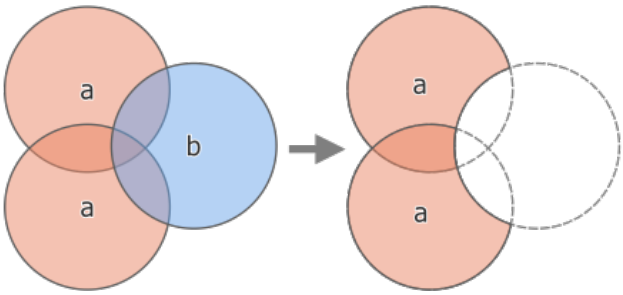
9.13 Fig. : Densify geometry at a given interval

Further reading: *Densify by interval* algorithm

difference

Returns a geometry that represents that part of geometry1 that does not intersect with geometry2.

Sintaksė	difference(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `geom_to_wkt( difference( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4)' ) ) )` → ‚LINESTRING(4 4, 5 5)‘



9.14 Fig. : Difference operation between a two-features input layer ‘a’ and a single feature overlay layer ‘b’ (left) - resulting in a new layer with the modified ‘a’ features (right)

Further reading: *Skirtumas* algorithm

disjoint

Tests whether geometries do not spatially intersect. Returns TRUE if the geometries do not share any space together.

Sintaksė	disjoint(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `disjoint( geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0 ))' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `disjoint( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'POINT(4 4)' ))` → FALSE

Further reading: *overlay_disjoint*

distance

Returns the minimum distance (based on spatial reference) between two geometries in projected units.

Sintaksė	distance(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `distance( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'POINT(4 8)' ) )` → 4

distance_to_vertex

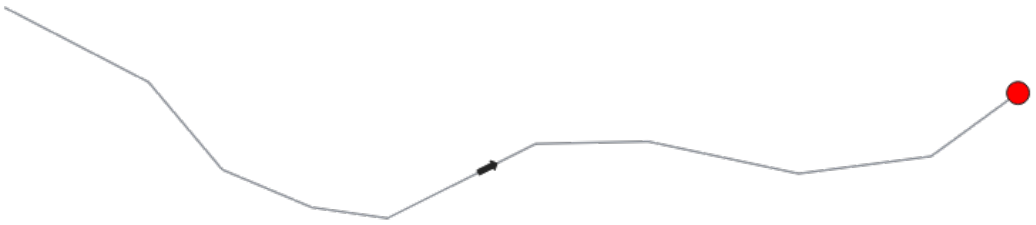
Returns the distance along the geometry to a specified vertex.

Sintaksė	distance_to_vertex(geometry, vertex)
Argumentai	• **geometry** - a linestring geometry • **vertex** - vertex index, starting from 0; if the value is negative, the selected vertex index will be its total count minus the absolute value
Pavyzdžiai	• `distance_to_vertex(geometry:=geom_from_wkt('LineString(0 0, 10 0, 10 10)'),vertex:=1)` → 10.0

end_point

Returns the last node from a geometry.

Sintaksė	end_point(geometry)
Argumentai	• **geometry** - geometry object
Pavyzdžiai	• `geom_to_wkt(end_point(geom_from_wkt('LINESTRING(4 0, 4 2, 0 2)')))` → ‚Point (0 2)‘



9.15 Fig. : End point of a line feature

Further reading: *start_point*, *Extract specific vertices* algorithm

exif_geotag

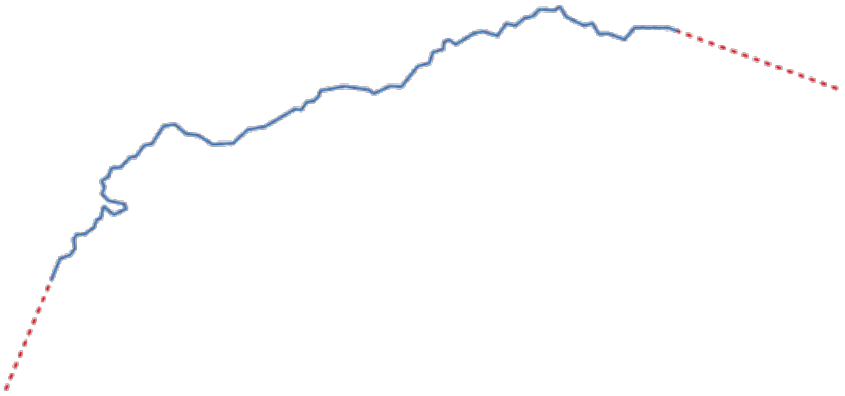
Creates a point geometry from the exif geotags of an image file.

Sintaksė	exif_geotag(path)
Argumentai	• **path** - An image file path or a map layer value. If a map layer value is specified then the file source of the layer will be used.
Pavyzdžiai	• `geom_to_wkt(exif_geotag('/my/photo.jpg'))` → ‚Point (2 4)‘

extend

Extends the start and end of a linestring geometry by a specified amount. Lines are extended using the bearing of the first and last segment in the line. For a multilinestring, all the parts are extended. Distances are in the Spatial Reference System of this geometry.

Sintaksė	extend(geometry, start_distance, end_distance)
Argumentai	• **geometry** - a (multi)linestring geometry • **start_distance** - distance to extend the start of the line • **end_distance** - distance to extend the end of the line.
Pavyzdžiai	• `geom_to_wkt(extend(geom_from_wkt('LineString(0 0, 1 0, 1 1)'),1,2))` → ‚LineString (-1 0, 1 0, 1 3)‘ • `geom_to_wkt(extend(geom_from_wkt('MultiLineString((0 0, 1 0, 1 1), (2 2, 0 2, 0 5))'),1,2))` → ‚MultiLineString ((-1 0, 1 0, 1 3),(3 2, 0 2, 0 7))‘



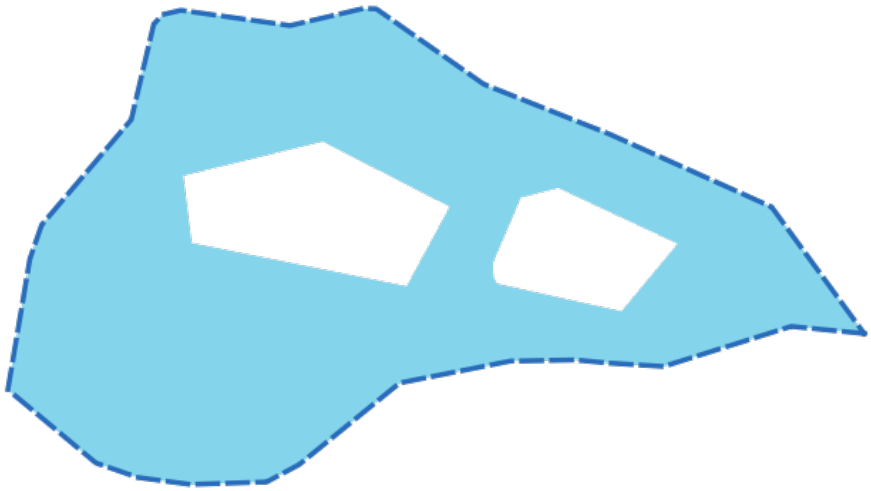
9.16 Fig. : The red dashes represent the initial and final extension of the original layer

Further reading: *Extend lines* algorithm

exterior_ring

Returns a line string representing the exterior ring of a polygon geometry. If the geometry is not a polygon then the result will be NULL.

Sintaksė	exterior_ring(geometry)
Argumentai	• **geometry** - a polygon geometry
Pavyzdžiai	• `geom_to_wkt(exterior_ring(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),( 0.1 0.1, 0.1 0.2, 0.2 0.2, 0.2, 0.1, 0.1 0.1))')))` → ‚LineString (-1 -1, 4 0, 4 2, 0 2, -1 -1)‘



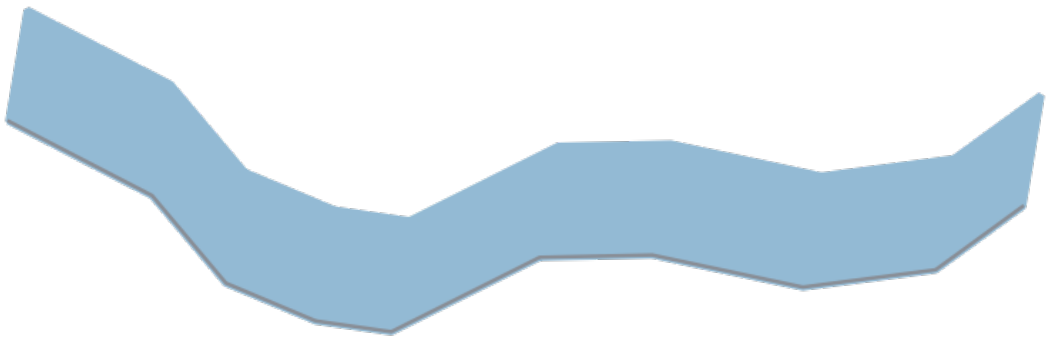
9.17 Fig. : The dashed line represents the exterior ring of the polygon

Further reading: *Ribos* algorithm, *interior_ring_n*

extrude

Returns an extruded version of the input (Multi-)Curve or (Multi-)Linestring geometry with an extension specified by x and y.

Sintaksė	extrude(geometry, x, y)
Argumentai	• **geometry** - a curve or linestring geometry • **x** - x extension, numeric value • **y** - y extension, numeric value
Pavyzdžiai	• `geom_to_wkt(extrude(geom_from_wkt('LineString(1 2, 3 2, 4 3)'), 1, 2))` → ‚Polygon ((1 2, 3 2, 4 3, 5 5, 4 4, 2 4, 1 2))‘ • `geom_to_wkt(extrude(geom_from_wkt('MultiLineString((1 2, 3 2), (4 3, 8 3))'), 1, 2))` → ‚MultiPolygon (((1 2, 3 2, 4 4, 2 4, 1 2)),((4 3, 8 3, 9 5, 5 5, 4 3)))‘



9.18 Fig. : Generating a polygon by extruding a line with offset in x and y directions

flip_coordinates

Returns a copy of the geometry with the x and y coordinates swapped. Useful for repairing geometries which have had their latitude and longitude values reversed.

Sintaksė	flip_coordinates(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `geom_to_wkt(flip_coordinates(make_point(1, 2)))` → ‚Point (2 1)‘ • `geom_to_wkt(flip_coordinates(geom_from_wkt('LineString(0 2, 1 0, 1 6)')))` → ‚LineString (2 0, 0 1, 6 1)‘

Further reading: *Swap X and Y coordinates* algorithm

force_polygon_ccw

Forces a geometry to respect the convention where exterior rings are counter-clockwise, interior rings are clockwise.

Sintaksė	force_polygon_ccw(geometry)
Argumentai	• **geometry** - a geometry. Any non-polygon geometries are returned unchanged.
Pavyzdžiai	• `geom_to_wkt(force_polygon_ccw(geometry:=geom_from_wkt('Polygon ((-1 -1, 0 2, 4 2, 4 0, -1 -1))')))` → ‚Polygon ((-1 -1, 4 0, 4 2, 0 2, -1 -1))‘

Further reading: *force_polygon_cw*, *force_rhr*

force_polygon_cw

Forces a geometry to respect the convention where exterior rings are clockwise, interior rings are counter-clockwise.

Sintaksė	force_polygon_cw(geometry)
Argumentai	• **geometry** - a geometry. Any non-polygon geometries are returned unchanged.
Pavyzdžiai	• `geom_to_wkt(force_polygon_cw(geometry:=geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))')))` → ‚Polygon ((-1 -1, 0 2, 4 2, 4 0, -1 -1))‘

Further reading: *force_polygon_ccw*, *force_rhr*

force_rhr

Forces a geometry to respect the Right-Hand-Rule, in which the area that is bounded by a polygon is to the right of the boundary. In particular, the exterior ring is oriented in a clockwise direction and the interior rings in a counter-clockwise direction. Due to the inconsistency in the definition of the Right-Hand-Rule in some contexts it is recommended to use the explicit force_polygon_cw function instead.

Sintaksė	force_rhr(geometry)
Argumentai	• **geometry** - a geometry. Any non-polygon geometries are returned unchanged.
Pavyzdžiai	• `geom_to_wkt(force_rhr(geometry:=geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))')))` → ‚Polygon ((-1 -1, 0 2, 4 2, 4 0, -1 -1))‘

Further reading: *Force right-hand-rule* algorithm, *force_polygon_ccw*, *force_polygon_cw*

geom_from_gml

Returns a geometry from a GML representation of geometry.

Sintaksė	geom_from_gml(gml)
Argumentai	• **gml** - GML representation of a geometry as a string
Pavyzdžiai	• `geom_from_gml('<gml:LineString srsName="EPSG:4326"><gml:coordinates>4, 4 5,5 6,6</gml:coordinates></gml:LineString>')` → a line geometry object

geom_from_wkb

Returns a geometry created from a Well-Known Binary (WKB) representation.

Sintaksė	geom_from_wkb(binary)
Argumentai	• **binary** - Well-Known Binary (WKB) representation of a geometry (as a binary blob)
Pavyzdžiai	• `geom_from_wkb( geom_to_wkb( make_point(4,5) ) )` → a point geometry object

geom_from_wkt

Returns a geometry created from a Well-Known Text (WKT) representation.

Sintaksė	geom_from_wkt(text)
Argumentai	• **text** - Well-Known Text (WKT) representation of a geometry
Pavyzdžiai	• `geom_from_wkt( 'POINT(4 5)' )` → a geometry object

geom_to_wkb

Returns the Well-Known Binary (WKB) representation of a geometry

Sintaksė	geom_to_wkb(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `geom_to_wkb( @geometry )` → binary blob containing a geometry object

geom_to_wkt

Returns the Well-Known Text (WKT) representation of the geometry without SRID metadata.

Sintaksė	geom_to_wkt(geometry, [precision=8]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **precision** - numeric precision
Pavyzdžiai	• `geom_to_wkt( make_point(6, 50) )` → ‚POINT(6 50)‘ • `geom_to_wkt(centroid(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')))` → ‚POINT(0 0.66666667)‘ • `geom_to_wkt(centroid(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')), 2)` → ‚POINT(0 0.67)‘

$geometry

Returns the geometry of the current feature. Can be used for processing with other functions. **WARNING: This function is deprecated. It is recommended to use the replacement @geometry variable instead.**

Sintaksė	$geometry
Pavyzdžiai	• `geom_to_wkt( $geometry )` → ‚POINT(6 50)‘

geometry

Returns a feature's geometry.

Sintaksė	geometry(feature)
Argumentai	• **feature** - a feature object
Pavyzdžiai	• `geometry( @feature )` → the geometry of the current feature. Prefer using @geometry. • `geom_to_wkt( geometry( get_feature_by_id( 'streets', 1 ) ) )` → the geometry in WKT of the feature with the id 1 on the layer „streets“, e.g. ‚POINT(6 50)‘ • `intersects( @geometry, geometry( get_feature( 'streets', 'name', 'Main St.' ) ) )` → TRUE if the current feature spatially intersects the ‚Main St.‘ named feature in the „streets“ layer

geometry_n

Returns a specific geometry from a geometry collection, or NULL if the input geometry is not a collection. Also returns a part from a multipart geometry.

Sintaksė	geometry_n(geometry, index)
Argumentai	• **geometry** - geometry collection • **index** - index of geometry to return, where 1 is the first geometry in the collection
Pavyzdžiai	• `geom_to_wkt(geometry_n(geom_from_wkt('GEOMETRYCOLLECTION(POINT(0 1), POINT(0 0), POINT(1 0), POINT(1 1))'),3))` → ‚Point (1 0)‘

geometry_type

Returns a string value describing the type of a geometry (Point, Line or Polygon)

Sintaksė	geometry_type(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `geometry_type( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → ‚Line‘ • `geometry_type( geom_from_wkt( 'MULTILINESTRING((2 5, 3 6, 4 8), (1 1, 0 0))') )` → ‚Line‘ • `geometry_type( geom_from_wkt( 'POINT(2 5)') )` → ‚Point‘ • `geometry_type( geom_from_wkt( 'POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))') )` → ‚Polygon‘

hausdorff_distance

Returns the Hausdorff distance between two geometries. This is basically a measure of how similar or dissimilar 2 geometries are, with a lower distance indicating more similar geometries.

The function can be executed with an optional densify fraction argument. If not specified, an approximation to the standard Hausdorff distance is used. This approximation is exact or close enough for a large subset of useful cases. Examples of these are:

- computing distance between Linestrings that are roughly parallel to each other, and roughly equal in length. This occurs in matching linear networks.
- Testing similarity of geometries.

If the default approximate provided by this method is insufficient, specify the optional densify fraction argument. Specifying this argument performs a segment densification before computing the discrete Hausdorff distance. The parameter sets the fraction by which to densify each segment. Each segment will be split into a number of equal-length subsegments, whose fraction of the total length is closest to the given fraction. Decreasing the densify fraction parameter will make the distance returned approach the true Hausdorff distance for the geometries.

Sintaksė	hausdorff_distance(geometry1, geometry2, [densify_fraction]) [] žymi neprivalomus argumentus
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry • **densify_fraction** - densify fraction amount
Pavyzdžiai	• `hausdorff_distance( geometry1:= geom_from_wkt('LINESTRING (0 0, 2 1)'),geometry2:=geom_from_wkt('LINESTRING (0 0, 2 0)'))` → 2 • `hausdorff_distance( geom_from_wkt('LINESTRING (130 0, 0 0, 0 150)'),geom_from_wkt('LINESTRING (10 10, 10 150, 130 10)'))` → 14.142135623 • `hausdorff_distance( geom_from_wkt('LINESTRING (130 0, 0 0, 0 150)'),geom_from_wkt('LINESTRING (10 10, 10 150, 130 10)'), 0.5)` → 70.0

inclination

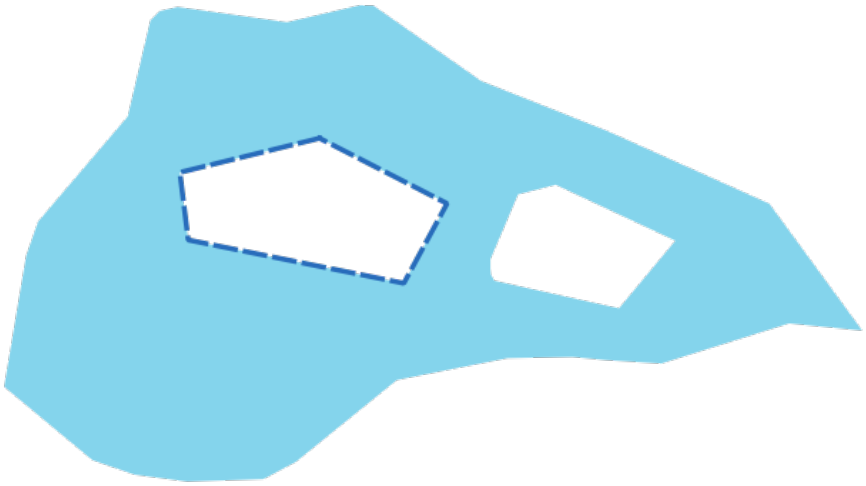
Returns the inclination measured from the zenith (0) to the nadir (180) on point_a to point_b.

Sintaksė	inclination(point_a, point_b)
Argumentai	• **point_a** - point geometry • **point_b** - point geometry
Pavyzdžiai	• `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, 5 ) )` → 0.0 • `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, 0 ) )` → 90.0 • `inclination( make_point( 5, 10, 0 ), make_point( 50, 100, 0 ) )` → 90.0 • `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, -5 ) )` → 180.0

interior_ring_n

Returns a specific interior ring from a polygon geometry, or NULL if the geometry is not a polygon.

Sintaksė	interior_ring_n(geometry, index)
Argumentai	• **geometry** - polygon geometry • **index** - index of interior to return, where 1 is the first interior ring
Pavyzdžiai	• `geom_to_wkt(interior_ring_n(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),(-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1),(-1 -1, 4 0, 4 2, 0 2, -1 -1))'),1))` → ‚LineString (-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1)‘



9.19 Fig. : The dashed line represents a specific interior ring of the polygon

Further reading: *exterior_ring*

intersection

Returns a geometry that represents the shared portion of two geometries.

Sintaksė	intersection(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `geom_to_wkt( intersection( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4)' ) ) )` → ‚LINESTRING(3 3, 4 4)‘ • `geom_to_wkt( intersection( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'MULTIPOINT(3.5 3.5, 4 5)' ) ) )` → ‚POINT(3.5 3.5)‘

Further reading: *Intersection* algorithm

intersects

Tests whether a geometry intersects another. Returns TRUE if the geometries spatially intersect (share any portion of space) and false if they do not.

Sintaksė	intersects(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `intersects( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `intersects( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'POINT(5 5)' ) )` → FALSE

Further reading: *overlay_intersects*

intersects_bbox

Tests whether a geometry's bounding box overlaps another geometry's bounding box. Returns TRUE if the geometries spatially intersect the bounding box defined and false if they do not.

Sintaksė	intersects_bbox(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `intersects_bbox( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `intersects_bbox( geom_from_wkt( 'POINT(6 5)' ), geom_from_wkt( 'POLYGON((3 3, 4 4, 5 5, 3 3))' ) )` → FALSE

is_closed

Returns TRUE if a line string is closed (start and end points are coincident), or false if a line string is not closed. If the geometry is not a line string then the result will be NULL.

Sintaksė	is_closed(geometry)
Argumentai	• **geometry** - a line string geometry
Pavyzdžiai	• `is_closed(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)'))` → FALSE • `is_closed(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2, 0 0)'))` → TRUE

is_empty

Returns TRUE if a geometry is empty (without coordinates), false if the geometry is not empty and NULL if there is no geometry. See also is_empty_or_null.

Sintaksė	is_empty(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `is_empty(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)'))` → FALSE • `is_empty(geom_from_wkt('LINESTRING EMPTY'))` → TRUE • `is_empty(geom_from_wkt('POINT(7 4)'))` → FALSE • `is_empty(geom_from_wkt('POINT EMPTY'))` → TRUE

Further reading: *is_empty_or_null*

is_empty_or_null

Returns TRUE if a geometry is NULL or empty (without coordinates) or false otherwise. This function is like the expression ‚@geometry IS NULL or is_empty(@geometry)‘

Sintaksė	is_empty_or_null(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `is_empty_or_null(NULL)` → TRUE • `is_empty_or_null(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)'))` → FALSE • `is_empty_or_null(geom_from_wkt('LINESTRING EMPTY'))` → TRUE • `is_empty_or_null(geom_from_wkt('POINT(7 4)'))` → FALSE • `is_empty_or_null(geom_from_wkt('POINT EMPTY'))` → TRUE

Further reading: *is_empty*, *NULL*

is_multipart

Returns TRUE if the geometry is of Multi type.

Sintaksė	is_multipart(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `is_multipart(geom_from_wkt('MULTIPOINT ((0 0),(1 1),(2 2))'))` → TRUE • `is_multipart(geom_from_wkt('POINT (0 0)'))` → FALSE

is_valid

Returns TRUE if a geometry is valid; if it is well-formed in 2D according to the OGC rules.

Sintaksė	is_valid(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `is_valid(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2, 0 0)'))` → TRUE • `is_valid(geom_from_wkt('LINESTRING(0 0)'))` → FALSE

Further reading: *make_valid*, *Tikrinti tvarkingumą* algorithm

$length

Returns the length of a linestring. If you need the length of a border of a polygon, use $perimeter instead. The length calculated by this function respects both the current project's ellipsoid setting and distance unit settings. For example, if an ellipsoid has been set for the project then the calculated length will be ellipsoidal, and if no ellipsoid is set then the calculated length will be planimetric.

Sintaksė	$length
Pavyzdžiai	• `$length` → 42.4711

length

Returns the number of characters in a string or the length of a geometry linestring.

String variant

Returns the number of characters in a string.

Sintaksė	length(string)
Argumentai	• **string** - string to count length of
Pavyzdžiai	• `length('hello')` → 5

Geometry variant

Calculate the length of a geometry line object. Calculations are always planimetric in the Spatial Reference System (SRS) of this geometry, and the units of the returned length will match the units for the SRS. This differs from the calculations performed by the $length function, which will perform ellipsoidal calculations based on the project's ellipsoid and distance unit settings.

Sintaksė	length(geometry)
Argumentai	• **geometry** - line geometry object
Pavyzdžiai	• `length(geom_from_wkt('LINESTRING(0 0, 4 0)'))` → 4.0

Further reading: *straight_distance_2d*

length3D

Calculates the 3D length of a geometry line object. If the geometry is not a 3D line object, it returns its 2D length. Calculations are always planimetric in the Spatial Reference System (SRS) of this geometry, and the units of the returned length will match the units for the SRS. This differs from the calculations performed by the $length function, which will perform ellipsoidal calculations based on the project's ellipsoid and distance unit settings.

Sintaksė	length3D(geometry)
Argumentai	• **geometry** - line geometry object
Pavyzdžiai	• `length3D(geom_from_wkt('LINESTRINGZ(0 0 0, 3 0 4)'))` → 5.0

line_interpolate_angle

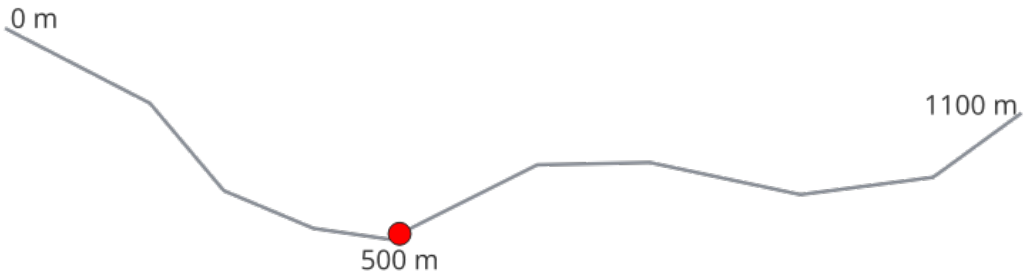
Returns the angle parallel to the geometry at a specified distance along a linestring geometry. Angles are in degrees clockwise from north.

Sintaksė	line_interpolate_angle(geometry, distance)
Argumentai	• **geometry** - a linestring geometry • **distance** - distance along line to interpolate angle at
Pavyzdžiai	• `line_interpolate_angle(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),distance:=5)` → 90.0

line_interpolate_point

Returns the point interpolated by a specified distance along a linestring geometry.

Sintaksė	line_interpolate_point(geometry, distance)
Argumentai	• **geometry** - a linestring geometry • **distance** - distance along line to interpolate
Pavyzdžiai	• `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 8 0)'), distance:=5))` → ‚Point (5 0)‘ • `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 1 1, 2 0)'), distance:=2.1))` → ‚Point (1.48492424 0.51507576)‘ • `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 1 0)'), distance:=2))` → NULL



9.20 Fig. : Interpolated point at 500m of the beginning of the line

Further reading: *Interpolate point on line* algorithm

line_interpolate_point_by_m

Returns the point interpolated by a matching M value along a linestring geometry.

Sintaksė	line_interpolate_point_by_m(geometry, m, [use_3d_distance=false]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a linestring geometry • **m** - an M value • **use_3d_distance** - controls whether 2D or 3D distances between vertices should be used during interpolation (this option is only considered for lines with z values)
Pavyzdžiai	• `geom_to_wkt(line_interpolate_point_by_m(geom_from_wkt('LineStringM(0 0 0, 10 10 10)'), m:=5))` → ‚Point (5 5)‘

line_locate_m

Returns the distance along a linestring corresponding to the first matching interpolated M value.

Sintaksė	line_locate_m(geometry, m, [use_3d_distance=false]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a linestring geometry • **m** - an M value • **use_3d_distance** - controls whether 2D or 3D distances between vertices should be used during interpolation (this option is only considered for lines with z values)
Pavyzdžiai	• `line_locate_m(geometry:=geom_from_wkt('LineStringM(0 0 0, 10 10 10)'),m:=5)` → 7.07106

line_locate_point

Returns the distance along a linestring corresponding to the closest position the linestring comes to a specified point geometry.

Sintaksė	line_locate_point(geometry, point)
Argumentai	• **geometry** - a linestring geometry • **point** - point geometry to locate closest position on linestring to
Pavyzdžiai	• `line_locate_point(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),point:=geom_from_wkt('Point(5 0)'))` → 5.0

line_merge

Returns a LineString or MultiLineString geometry, where any connected LineStrings from the input geometry have been merged into a single linestring. This function will return NULL if passed a geometry which is not a LineString/MultiLineString.

Sintaksė	line_merge(geometry)
Argumentai	• **geometry** - a LineString/MultiLineString geometry
Pavyzdžiai	• `geom_to_wkt(line_merge(geom_from_wkt('MULTILINESTRING((0 0, 1 1),(1 1, 2 2))')))` → ‚LineString(0 0,1 1,2 2)‘ • `geom_to_wkt(line_merge(geom_from_wkt('MULTILINESTRING((0 0, 1 1),(11 1, 21 2))')))` → ‚MultiLineString((0 0, 1 1),(11 1, 21 2)‘

line_substring

Returns the portion of a line (or curve) geometry which falls between the specified start and end distances (measured from the beginning of the line). Z and M values are linearly interpolated from existing values.

Sintaksė	line_substring(geometry, start_distance, end_distance)
Argumentai	• **geometry** - a linestring or curve geometry • **start_distance** - distance to start of substring • **end_distance** - distance to end of substring
Pavyzdžiai	• `geom_to_wkt(line_substring(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),start_distance:=2,end_distance:=6))` → ‚LineString (2 0,6 0)‘

9.21 Fig. : Substring line with starting distance set at 0 meters and the ending distance at 250 meters.

Further reading: *Line substring* algorithm

m

Returns the m (measure) value of a point geometry.

Sintaksė	m(geometry)
Argumentai	• **geometry** - a point geometry
Pavyzdžiai	• `m( geom_from_wkt( 'POINTM(2 5 4)' ) )` → 4

m_at

Retrieves a m coordinate of the geometry, or NULL if the geometry has no m value.

Sintaksė	m_at(geometry, vertex)
Argumentai	• **geometry** - geometry object • **vertex** - index of the vertex of the geometry (indices start at 0; negative values apply from the last index, starting at -1)
Pavyzdžiai	• `m_at(geom_from_wkt('LineStringZM(0 0 0 0, 10 10 0 5, 10 10 0 0)'), 1)` → 5

m_max

Returns the maximum m (measure) value of a geometry.

Sintaksė	m_max(geometry)
Argumentai	• **geometry** - a geometry containing m values
Pavyzdžiai	• `m_max( make_point_m( 0,0,1 ) )` → 1 • `m_max(make_line( make_point_m( 0,0,1 ), make_point_m( -1,-1,2 ), make_point_m( -2,-2,0 ) ) )` → 2

m_min

Returns the minimum m (measure) value of a geometry.

Sintaksė	m_min(geometry)
Argumentai	• **geometry** - a geometry containing m values
Pavyzdžiai	• `m_min( make_point_m( 0,0,1 ) )` → 1 • `m_min(make_line( make_point_m( 0,0,1 ), make_point_m( -1, -1,2 ), make_point_m( -2,-2,0 ) ) )` → 0

main_angle

Returns the angle of the long axis (clockwise, in degrees from North) of the oriented minimal bounding rectangle, which completely covers the geometry.

Sintaksė	main_angle(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `main_angle(geom_from_wkt('Polygon ((321577 129614, 321581 129618, 321585 129615, 321581 129610, 321577 129614))'))` → 38.66

make_circle

Creates a circular polygon.

Sintaksė	make_circle(center, radius, [segments=36]) [] žymi neprivalomus argumentus
Argumentai	• **center** - center point of the circle • **radius** - radius of the circle • **segments** - optional argument for polygon segmentation. By default this value is 36
Pavyzdžiai	• `geom_to_wkt(make_circle(make_point(10,10), 5, 4))` → ‚Polygon ((10 15, 15 10, 10 5, 5 10, 10 15))‘ • `geom_to_wkt(make_circle(make_point(10,10,5), 5, 4))` → ‚PolygonZ ((10 15 5, 15 10 5, 10 5 5, 5 10 5, 10 15 5))‘ • `geom_to_wkt(make_circle(make_point(10,10,5,30), 5, 4))` → ‚PolygonZM ((10 15 5 30, 15 10 5 30, 10 5 5 30, 5 10 5 30, 10 15 5 30))‘

make_ellipse

Creates an elliptical polygon.

Sintaksė	make_ellipse(center, semi_major_axis, semi_minor_axis, azimuth, [segments=36]) [] žymi neprivalomus argumentus
Argumentai	• **center** - center point of the ellipse • **semi_major_axis** - semi-major axis of the ellipse • **semi_minor_axis** - semi-minor axis of the ellipse • **azimuth** - orientation of the ellipse • **segments** - optional argument for polygon segmentation. By default this value is 36
Pavyzdžiai	• `geom_to_wkt(make_ellipse(make_point(10,10), 5, 2, 90, 4))` → ‚Polygon ((15 10, 10 8, 5 10, 10 12, 15 10))‘ • `geom_to_wkt(make_ellipse(make_point(10,10,5), 5, 2, 90, 4))` → ‚PolygonZ ((15 10 5, 10 8 5, 5 10 5, 10 12 5, 15 10 5))‘ • `geom_to_wkt(make_ellipse(make_point(10,10,5,30), 5, 2, 90, 4))` → ‚PolygonZM ((15 10 5 30, 10 8 5 30, 5 10 5 30, 10 12 5 30, 15 10 5 30))‘

make_line

Creates a line geometry from a series of point geometries.

List of arguments variant

Line vertices are specified as separate arguments to the function.

Sintaksė	make_line(point1, point2, …)
Argumentai	• **point** - a point geometry (or array of points)
Pavyzdžiai	• `geom_to_wkt(make_line(make_point(2,4),make_point(3,5)))` → ‚LineString (2 4, 3 5)‘ • `geom_to_wkt(make_line(make_point(2,4),make_point(3,5), make_point(9,7)))` → ‚LineString (2 4, 3 5, 9 7)‘

Array variant

Line vertices are specified as an array of points.

Sintaksė	make_line(array)
Argumentai	• **array** - array of points
Pavyzdžiai	• `geom_to_wkt(make_line(array(make_point(2,4),make_point(3,5),make_point(9,7))))` → ‚LineString (2 4, 3 5, 9 7)‘

make_point

Creates a point geometry from an x and y (and optional z and m) value.

Sintaksė	make_point(x, y, [z], [m]) [] žymi neprivalomus argumentus
Argumentai	• **x** - x coordinate of point • **y** - y coordinate of point • **z** - optional z coordinate of point • **m** - optional m value of point
Pavyzdžiai	• `geom_to_wkt(make_point(2,4))` → ‚Point (2 4)‘ • `geom_to_wkt(make_point(2,4,6))` → ‚PointZ (2 4 6)‘ • `geom_to_wkt(make_point(2,4,6,8))` → ‚PointZM (2 4 6 8)‘

make_point_m

Creates a point geometry from an x, y coordinate and m value.

Sintaksė	make_point_m(x, y, m)
Argumentai	• **x** - x coordinate of point • **y** - y coordinate of point • **m** - m value of point
Pavyzdžiai	• `geom_to_wkt(make_point_m(2,4,6))` → ‚PointM (2 4 6)‘

make_polygon

Creates a polygon geometry from an outer ring and optional series of inner ring geometries.

Sintaksė	make_polygon(outerRing, [innerRing1], [innerRing2], …) [] žymi neprivalomus argumentus
Argumentai	• **outerRing** - closed line geometry for polygon’s outer ring • **innerRing** - optional closed line geometry for inner ring
Pavyzdžiai	• `geom_to_wkt(make_polygon(geom_from_wkt('LINESTRING( 0 0, 0 1, 1 1, 1 0, 0 0 )')))` → ‚Polygon ((0 0, 0 1, 1 1, 1 0, 0 0))‘ • `geom_to_wkt(make_polygon(geom_from_wkt('LINESTRING( 0 0, 0 1, 1 1, 1 0, 0 0 )'),geom_from_wkt('LINESTRING( 0.1 0.1, 0.1 0.2, 0.2 0.2, 0.2 0.1, 0.1 0.1 )'), geom_from_wkt('LINESTRING( 0.8 0.8, 0.8 0.9, 0.9 0.9, 0.9 0.8, 0.8 0.8 )')))` → ‚Polygon ((0 0, 0 1, 1 1, 1 0, 0 0),(0.1 0.1, 0.1 0.2, 0.2 0.2, 0.2 0.1, 0.1 0.1),(0.8 0.8, 0.8 0.9, 0.9 0.9, 0.9 0.8, 0.8 0.8))‘

make_rectangle_3points

Creates a rectangle from 3 points.

Sintaksė	make_rectangle_3points(point1, point2, point3, [option=0]) [] žymi neprivalomus argumentus
Argumentai	• **point1** - First point. • **point2** - Second point. • **point3** - Third point. • **option** - An optional argument to construct the rectangle. By default this value is 0. Value can be 0 (distance) or 1 (projected). Option distance: Second distance is equal to the distance between 2nd and 3rd point. Option projected: Second distance is equal to the distance of the perpendicular projection of the 3rd point on the segment or its extension.
Pavyzdžiai	• `geom_to_wkt(make_rectangle_3points(make_point(0, 0), make_point(0,5), make_point(5, 5), 0))` → ‚Polygon ((0 0, 0 5, 5 5, 5 0, 0 0))‘ • `geom_to_wkt(make_rectangle_3points(make_point(0, 0), make_point(0,5), make_point(5, 3), 1))` → ‚Polygon ((0 0, 0 5, 5 5, 5 0, 0 0))‘

make_regular_polygon

Creates a regular polygon.

Sintaksė	make_regular_polygon(center, radius, number_sides, [circle=0]) [] žymi neprivalomus argumentus
Argumentai	• **center** - center of the regular polygon • **radius** - second point. The first if the regular polygon is inscribed. The midpoint of the first side if the regular polygon is circumscribed. • **number_sides** - Number of sides/edges of the regular polygon • **circle** - Optional argument to construct the regular polygon. By default this value is 0. Value can be 0 (inscribed) or 1 (circumscribed)
Pavyzdžiai	• `geom_to_wkt(make_regular_polygon(make_point(0,0), make_point(0,5), 5))` → ‚Polygon ((0 5, 4.76 1.55, 2.94 -4.05, -2.94 -4.05, -4.76 1.55, 0 5))‘ • `geom_to_wkt(make_regular_polygon(make_point(0,0), project(make_point(0,0), 4.0451, radians(36)), 5))` → ‚Polygon ((0 5, 4.76 1.55, 2.94 -4.05, -2.94 -4.05, -4.76 1.55, 0 5))‘

make_square

Creates a square from a diagonal.

Sintaksė	make_square(point1, point2)
Argumentai	• **point1** - First point of the diagonal • **point2** - Last point of the diagonal
Pavyzdžiai	• `geom_to_wkt(make_square( make_point(0,0), make_point(5, 5)))` → ‚Polygon ((0 0, -0 5, 5 5, 5 0, 0 0))‘ • `geom_to_wkt(make_square( make_point(5,0), make_point(5, 5)))` → ‚Polygon ((5 0, 2.5 2.5, 5 5, 7.5 2.5, 5 0))‘

make_triangle

Creates a triangle polygon.

Sintaksė	make_triangle(point1, point2, point3)
Argumentai	• **point1** - first point of the triangle • **point2** - second point of the triangle • **point3** - third point of the triangle
Pavyzdžiai	• `geom_to_wkt(make_triangle(make_point(0,0), make_point(5, 5), make_point(0,10)))` → ‚Triangle ((0 0, 5 5, 0 10, 0 0))‘ • `geom_to_wkt(boundary(make_triangle(make_point(0,0), make_point(5,5), make_point(0,10))))` → ‚LineString (0 0, 5 5, 0 10, 0 0)‘

make_valid

Returns a valid geometry or an empty geometry if the geometry could not be made valid.

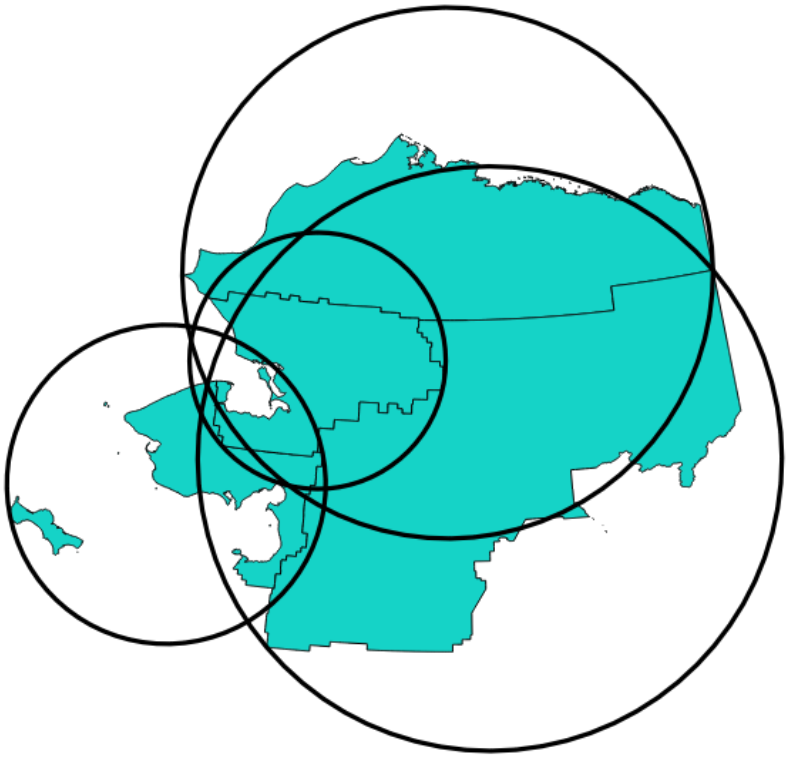
Sintaksė	make_valid(geometry, [method=structure], [keep_collapsed=false]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **method** - repair algorithm. May be either ‚structure' or ‚linework'. The ‚linework' option combines all rings into a set of noded lines and then extracts valid polygons from that linework. The ‚structure' method first makes all rings valid and then merges shells and subtracts holes from shells to generate valid result. Assumes that holes and shells are correctly categorized. • **keep_collapsed** - if set to true, then components that have collapsed into a lower dimensionality will be kept. For example, a ring collapsing to a line, or a line collapsing to a point.
Pavyzdžiai	• `geom_to_wkt(make_valid(geom_from_wkt('POLYGON((3 2, 4 1, 5 8, 3 2, 4 2))')))` → ‚Polygon ((3 2, 5 8, 4 1, 3 2))' • `geom_to_wkt(make_valid(geom_from_wkt('POLYGON((3 2, 4 1, 5 8, 3 2, 4 2))'), 'linework'))` → ‚GeometryCollection (Polygon ((5 8, 4 1, 3 2, 5 8)),LineString (3 2, 4 2))' • `geom_to_wkt(make_valid(geom_from_wkt('POLYGON((3 2, 4 1, 5 8))'), method:='linework'))` → ‚Polygon ((3 2, 4 1, 5 8, 3 2))' • `make_valid(geom_from_wkt('LINESTRING(0 0)'))` → An empty geometry

Further reading: *is_valid*, *Fix geometries* algorithm

minimal_circle

Returns the minimal enclosing circle of a geometry. It represents the minimum circle that encloses all geometries within the set.

Sintaksė	minimal_circle(geometry, [segments=36]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **segments** - optional argument for polygon segmentation. By default this value is 36
Pavyzdžiai	• `geom_to_wkt( minimal_circle( geom_from_wkt( 'LINESTRING(0 5, 0 -5, 2 1)' ), 4 ) )` → ‚Polygon ((0 5, 5 -0, -0 -5, -5 0, 0 5))' • `geom_to_wkt( minimal_circle( geom_from_wkt( 'MULTIPOINT(1 2, 3 4, 3 2)' ), 4 ) )` → ‚Polygon ((3 4, 3 2, 1 2, 1 4, 3 4))'



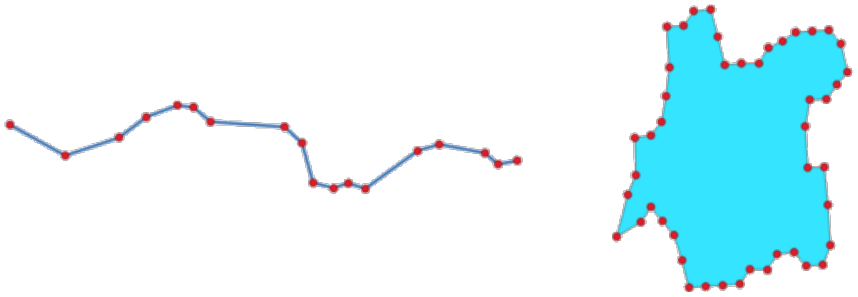
9.22 Fig. : Minimal enclosing circle of each feature

Further reading: *Minimum enclosing circles* algorithm

nodes_to_points

Returns a multipoint geometry consisting of every node in the input geometry.

Sintaksė	nodes_to_points(geometry, [ignore_closing_nodes=false]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - geometry object • **ignore_closing_nodes** - optional argument specifying whether to include duplicate nodes which close lines or polygons rings. Defaults to false, set to true to avoid including these duplicate nodes in the output collection.
Pavyzdžiai	• `geom_to_wkt(nodes_to_points(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → ‚MultiPoint ((0 0),(1 1),(2 2))‘ • `geom_to_wkt(nodes_to_points(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))'),true))` → ‚MultiPoint ((-1 -1),(4 0),(4 2),(0 2))‘



9.23 Fig. : Multi-point feature extracted from vertices

Further reading: *Extract vertices* algorithm

num_geometries

Returns the number of geometries in a geometry collection, or the number of parts in a multi-part geometry. The function returns NULL if the input geometry is not a collection.

Sintaksė	num_geometries(geometry)
Argumentai	• **geometry** - geometry collection or multi-part geometry
Pavyzdžiai	• `num_geometries(geom_from_wkt('GEOMETRYCOLLECTION(POINT(0 1), POINT(0 0), POINT(1 0), POINT(1 1))'))` → 4 • `num_geometries(geom_from_wkt('MULTIPOINT((0 1), (0 0), (1 0))'))` → 3

num_interior_rings

Returns the number of interior rings in a polygon or geometry collection, or NULL if the input geometry is not a polygon or collection.

Sintaksė	num_interior_rings(geometry)
Argumentai	• **geometry** - input geometry
Pavyzdžiai	• `num_interior_rings(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),(-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1))'))` → 1

num_points

Returns the number of vertices in a geometry.

Sintaksė	num_points(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `num_points(@geometry)` → number of vertices in the current feature's geometry

num_rings

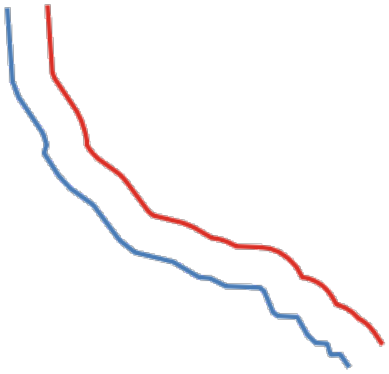
Returns the number of rings (including exterior rings) in a polygon or geometry collection, or NULL if the input geometry is not a polygon or collection.

Sintaksė	num_rings(geometry)
Argumentai	• **geometry** - input geometry
Pavyzdžiai	• `num_rings(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),(-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1))'))` → 2

offset_curve

Returns a geometry formed by offsetting a linestring geometry to the side. Distances are in the Spatial Reference System of this geometry.

Sintaksė	offset_curve(geometry, distance, [segments=8], [join=1], [miter_limit=2.0]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a (multi)linestring geometry • **distance** - offset distance. Positive values will be buffered to the left of lines, negative values to the right • **segments** - number of segments to use to represent a quarter circle when a round join style is used. A larger number results in a smoother line with more nodes. • **join** - join style for corners, where 1 = round, 2 = miter and 3 = bevel • **miter_limit** - limit on the miter ratio used for very sharp corners (when using miter joins only)
Pavyzdžiai	• `offset_curve(@geometry, 10.5)` → line offset to the left by 10.5 units • `offset_curve(@geometry, -10.5)` → line offset to the right by 10.5 units • `offset_curve(@geometry, 10.5, segments:=16, join:=1)` → line offset to the left by 10.5 units, using more segments to result in a smoother curve • `offset_curve(@geometry, 10.5, join:=3)` → line offset to the left by 10.5 units, using a beveled join



9.24 Fig. : Mėlynas - pradinis sluoksnis, raudonas - paslinktasis

Further reading: *Offset lines* algorithm

order_parts

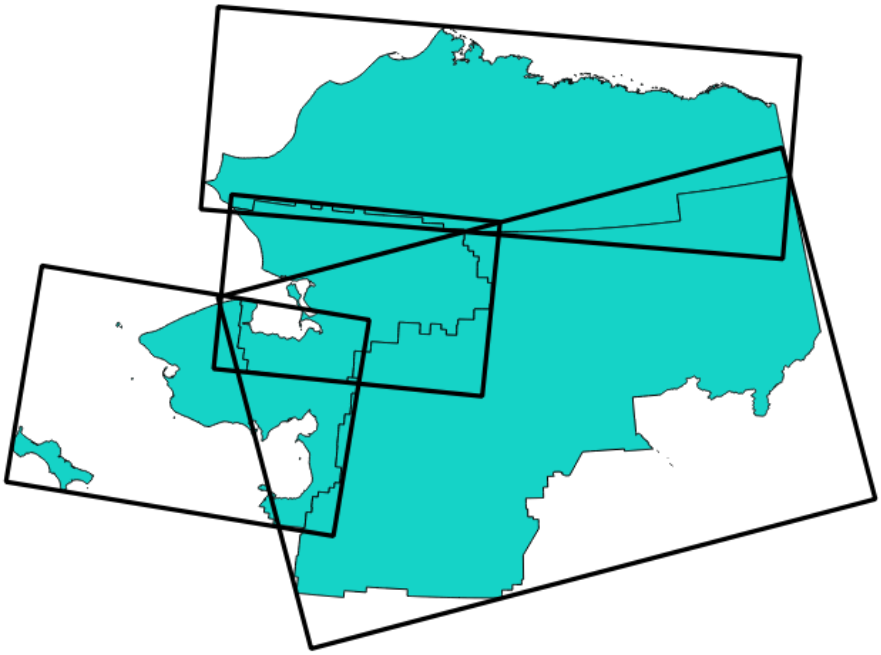
Orders the parts of a MultiGeometry by a given criteria

Sintaksė	order_parts(geometry, orderby, [ascending=true]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a multi-type geometry • **orderby** - an expression string defining the order criteria • **ascending** - boolean, True for ascending, False for descending
Pavyzdžiai	• `geom_to_wkt(order_parts(geom_from_wkt('MultiPolygon (((1 1, 5 1, 5 5, 1 5, 1 1)),((1 1, 9 1, 9 9, 1 9, 1 1)))'), 'area(@geometry)', False))` → ‚MultiPolygon (((1 1, 9 1, 9 9, 1 9, 1 1)),((1 1, 5 1, 5 5, 1 5, 1 1)))‘ • `geom_to_wkt(order_parts(geom_from_wkt('LineString(1 2, 3 2, 4 3)'), '1', True))` → ‚LineString(1 2, 3 2, 4 3)‘

oriented_bbox

Returns a geometry which represents the minimal oriented bounding box of an input geometry.

Sintaksė	oriented_bbox(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `geom_to_wkt( oriented_bbox( geom_from_wkt( 'MULTIPOINT(1 2, 3 4, 3 2)' ) ) )` → ‚Polygon ((3 2, 3 4, 1 4, 1 2, 3 2))‘



9.25 Fig. : Oriented minimum bounding box

Further reading: *Oriented minimum bounding box* algorithm

overlaps

Tests whether a geometry overlaps another. Returns TRUE if the geometries share space, are of the same dimension, but are not completely contained by each other.

Sintaksė	overlaps(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `overlaps( geom_from_wkt( 'LINESTRING(3 5, 4 4, 5 5, 5 3)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `overlaps( geom_from_wkt( 'LINESTRING(0 0, 1 1)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → FALSE

overlay_contains

Returns whether the current feature spatially contains at least one feature from a target layer, or an array of expression-based results for the features in the target layer contained in the current feature.

Read more on the underlying GEOS „Contains“ predicate, as described in PostGIS ST_Contains function.

Sintaksė	overlay_contains(layer, [expression], [filter], [limit], [cache=false]) [] žymi neprivalomus argumentus
Argumentai	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Pavyzdžiai	• `overlay_contains('regions')` → TRUE if the current feature spatially contains a region • `overlay_contains('regions', filter:= population > 10000)` → TRUE if the current feature spatially contains a region with a population greater than 10000 • `overlay_contains('regions', name)` → an array of names, for the regions contained in the current feature • `array_to_string(overlay_contains('regions', name))` → a string as a comma separated list of names, for the regions contained in the current feature • `array_sort(overlay_contains(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions contained in the current feature and with a population greater than 10000 • `overlay_contains(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions contained in the current feature

Further reading: *contains*, *array manipulation*, *Select by location* algorithm

overlay_crosses

Returns whether the current feature spatially crosses at least one feature from a target layer, or an array of expression-based results for the features in the target layer crossed by the current feature.

Read more on the underlying GEOS „Crosses“ predicate, as described in PostGIS ST_Crosses function.

Sintaksė	overlay_crosses(layer, [expression], [filter], [limit], [cache=false]) [] žymi neprivalomus argumentus
Argumentai	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Pavyzdžiai	• `overlay_crosses('regions')` → TRUE if the current feature spatially crosses a region • `overlay_crosses('regions', filter:= population > 10000)` → TRUE if the current feature spatially crosses a region with a population greater than 10000 • `overlay_crosses('regions', name)` → an array of names, for the regions crossed by the current feature • `array_to_string(overlay_crosses('regions', name))` → a string as a comma separated list of names, for the regions crossed by the current feature • `array_sort(overlay_crosses(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions crossed by the current feature and with a population greater than 10000 • `overlay_crosses(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions crossed by the current feature

Further reading: *crosses*, *array manipulation*, *Select by location* algorithm

overlay_disjoint

Returns whether the current feature is spatially disjoint from all the features of a target layer, or an array of expression-based results for the features in the target layer that are disjoint from the current feature.

Read more on the underlying GEOS „Disjoint“ predicate, as described in PostGIS ST_Disjoint function.

Sintaksė	overlay_disjoint(layer, [expression], [filter], [limit], [cache=false]) [] žymi neprivalomus argumentus
Argumentai	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Pavyzdžiai	• `overlay_disjoint('regions')` → TRUE if the current feature is spatially disjoint from all the regions • `overlay_disjoint('regions', filter:= population > 10000)` → TRUE if the current feature is spatially disjoint from all the regions with a population greater than 10000 • `overlay_disjoint('regions', name)` → an array of names, for the regions spatially disjoint from the current feature • `array_to_string(overlay_disjoint('regions', name))` → a string as a comma separated list of names, for the regions spatially disjoint from the current feature • `array_sort(overlay_disjoint(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions spatially disjoint from the current feature and with a population greater than 10000 • `overlay_disjoint(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions spatially disjoint from the current feature

Further reading: *disjoint*, *array manipulation*, *Select by location* algorithm

overlay_equals

Returns whether the current feature's geometry is exactly equal to at least one feature's geometry from a target layer, or an array of expression-based results for the features in the target layer whose geometry is exactly equal to the current feature's geometry. Note that the order of vertices matters.

Sintaksė	overlay_equals(layer, [expression], [filter], [limit], [cache=false]) [] žymi neprivalomus argumentus
Argumentai	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Pavyzdžiai	• `overlay_equals('regions')` → TRUE if the current feature's geometry is exactly the same as the geometry of a region • `overlay_equals('regions', filter:= population > 10000)` → TRUE if the current feature's geometry is exactly the same as the geometry of a region whose population is greater than 10000 • `overlay_equals('regions', name)` → an array of names, for the regions exactly equal to the current feature • `array_to_string(overlay_equals('regions', name))` → a string as a comma separated list of names, for the regions exactly equal to the current feature • `array_sort(overlay_equals(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions exactly equal to the current feature and with a population greater than 10000 • `overlay_equals(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions exactly equal to the current feature

Pastaba

This function requires exact equality of geometries, meaning that the geometries ‚LINESTRING(0 0, 1 1)‘ and ‚LINESTRING(1 1, 0 0)‘ are different.

Further reading: *array manipulation*, *Select by location* algorithm

overlay_intersects

Returns whether the current feature spatially intersects at least one feature from a target layer, or an array of expression-based results for the features in the target layer intersected by the current feature.

Read more on the underlying GEOS „Intersects“ predicate, as described in PostGIS ST_Intersects function.

Sintaksė	overlay_intersects(layer, [expression], [filter], [limit], [cache=false], [min_overlap], [min_inscribed_circle_radius], [return_details], [sort_by_intersection_size]) [] žymi neprivalomus argumentus
Argumentai	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider) • **min_overlap** - defines an optional exclusion filter: – for polygons, a minimum area in current feature squared units for the intersection. If the intersection results in multiple polygons the intersection will be returned if at least one of the polygons has an area greater or equal to the value – for lines, a minimum length in current feature units. If the intersection results in multiple lines the intersection will be returned if at least one of the lines has a length greater or equal to the value. • **min_inscribed_circle_radius** - defines an optional exclusion filter (for polygons only): minimum radius in current feature units for the maximum inscribed circle of the intersection. If the intersection results in multiple polygons the intersection will be returned if at least one of the polygons has a radius for the maximum inscribed circle greater or equal to the value. Read more on the underlying GEOS predicate, as described in PostGIS ST_MaximumInscribedCircle function. This argument requires GEOS >= 3.9. • **return_details** - Set this to true to return a list of maps containing (key names in quotes) the feature ‚id', the expression ‚result' and the ‚overlap' value (of the largest element in case of multipart). The ‚radius' of the maximum inscribed circle is also returned when the target layer is a polygon. Only valid when used with the expression parameter • **sort_by_intersection_size** - only valid when used with an expression, set this to ‚des' to return the results ordered by the overlap value in descending order or set this to ‚asc' for ascending order.
Pavyzdžiai	• `overlay_intersects('regions')` → TRUE if the current feature spatially intersects a region • `overlay_intersects('regions', filter:= population > 10000)` → TRUE if the current feature spatially intersects a region with a population greater than 10000 • `overlay_intersects('regions', name)` → an array of names, for the regions intersected by the current feature • `array_to_string(overlay_intersects('regions', name))` → a string as a comma separated list of names, for the regions intersected by the current feature • `array_sort(overlay_intersects(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions intersected by the current feature and with a population greater than 10000 • `overlay_intersects(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions intersected by the current feature • `overlay_intersects(layer:='regions', min_overlap:=0.54)` → TRUE if the current feature spatially intersects a region and the intersection area (of at least one of the parts in case of multipolygons) is greater or equal to 0.54 • `overlay_intersects(layer:='regions', min_inscribed_circle_radius:=0.54)` → TRUE if the current feature spatially intersects a region and the intersection area maximum inscribed circle's radius (of at least one of the parts in case of multipart) is greater or equal to 0.54 • `overlay_intersects(layer:='regions', expression:= geom_to_wkt(@geometry), return_details:=true)` → an array of maps containing ‚id', ‚result', ‚overlap' and ‚radius' • `overlay_intersects(layer:='regions', expression:=`

Further reading: *intersects*, *array manipulation*, *Select by location* algorithm

overlay_nearest

Returns whether the current feature has feature(s) from a target layer within a given distance, or an array of expression-based results for the features in the target layer within a distance from the current feature.

Note: This function can be slow and consume a lot of memory for large layers.

Sintaksė	overlay_nearest(layer, [expression], [filter], [limit=1], [max_distance], [cache=false]) [] žymi neprivalomus argumentus
Argumentai	• **layer** - the target layer • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features in the target layer will be used. • **limit** - an optional integer to limit the number of matching features. If not set, only the nearest feature will be returned. If set to -1, returns all the matching features. • **max_distance** - an optional distance to limit the search of matching features. If not set, all the features in the target layer will be used. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Pavyzdžiai	• `overlay_nearest('airports')` → TRUE if the „airports“ layer has at least one feature • `overlay_nearest('airports', max_distance:= 5000)` → TRUE if there is an airport within a distance of 5000 map units from the current feature • `overlay_nearest('airports', name)` → the name of the closest airport to the current feature, as an array • `array_to_string(overlay_nearest('airports', name))` → the name of the closest airport to the current feature, as a string • `overlay_nearest(layer:='airports', expression:= name, max_distance:= 5000)` → the name of the closest airport within a distance of 5000 map units from the current feature, as an array • `overlay_nearest(layer:='airports', expression:="name", filter:= "Use"='Civilian', limit:=3)` → an array of names, for up to the three closest civilian airports ordered by distance • `overlay_nearest(layer:='airports', expression:="name", limit:= -1, max_distance:= 5000)` → an array of names, for all the airports within a distance of 5000 map units from the current feature, ordered by distance

Further reading: *array manipulation*, *Join attributes by nearest* algorithm

overlay_touches

Returns whether the current feature spatially touches at least one feature from a target layer, or an array of expression-based results for the features in the target layer touched by the current feature.

Read more on the underlying GEOS „Touches“ predicate, as described in PostGIS ST_Touches function.

Sintaksė	overlay_touches(layer, [expression], [filter], [limit], [cache=false]) [] žymi neprivalomus argumentus
Argumentai	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Pavyzdžiai	• `overlay_touches('regions')` → TRUE if the current feature spatially touches a region • `overlay_touches('regions', filter:= population > 10000)` → TRUE if the current feature spatially touches a region with a population greater than 10000 • `overlay_touches('regions', name)` → an array of names, for the regions touched by the current feature • `array_to_string(overlay_touches('regions', name))` → a string as a comma separated list of names, for the regions touched by the current feature • `array_sort(overlay_touches(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions touched by the current feature and with a population greater than 10000 • `overlay_touches(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions touched by the current feature

Further reading: *touches*, *array manipulation*, *Select by location* algorithm

overlay_within

Returns whether the current feature is spatially within at least one feature from a target layer, or an array of expression-based results for the features in the target layer that contain the current feature.

Read more on the underlying GEOS „Within“ predicate, as described in PostGIS ST_Within function.

Sintaksė	overlay_within(layer, [expression], [filter], [limit], [cache=false]) [] žymi neprivalomus argumentus
Argumentai	• **layer** - the layer whose overlay is checked • **expression** - an optional expression to evaluate on the features from the target layer. If not set, the function will just return a boolean indicating whether there is at least one match. • **filter** - an optional expression to filter the target features to check. If not set, all the features will be checked. • **limit** - an optional integer to limit the number of matching features. If not set, all the matching features will be returned. • **cache** - set this to true to build a local spatial index (most of the time, this is unwanted, unless you are working with a particularly slow data provider)
Pavyzdžiai	• `overlay_within('regions')` → TRUE if the current feature is spatially within a region • `overlay_within('regions', filter:= population > 10000)` → TRUE if the current feature is spatially within a region with a population greater than 10000 • `overlay_within('regions', name)` → an array of names, for the regions containing the current feature • `array_to_string(overlay_within('regions', name))` → a string as a comma separated list of names, for the regions containing the current feature • `array_sort(overlay_within(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions containing the current feature and with a population greater than 10000 • `overlay_within(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions containing the current feature

Further reading: *within*, *array manipulation*, *Select by location* algorithm

$perimeter

Returns the perimeter length of the current feature. The perimeter calculated by this function respects both the current project's ellipsoid setting and distance unit settings. For example, if an ellipsoid has been set for the project then the calculated perimeter will be ellipsoidal, and if no ellipsoid is set then the calculated perimeter will be planimetric.

Sintaksė	$perimeter
Pavyzdžiai	• `$perimeter` → 42

perimeter

Returns the perimeter of a geometry polygon object. Calculations are always planimetric in the Spatial Reference System (SRS) of this geometry, and the units of the returned perimeter will match the units for the SRS. This differs from the calculations performed by the $perimeter function, which will perform ellipsoidal calculations based on the project's ellipsoid and distance unit settings.

Sintaksė	perimeter(geometry)
Argumentai	• **geometry** - polygon geometry object
Pavyzdžiai	• `perimeter(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'))` → 12.0

point_n

Returns a specific node from a geometry.

Sintaksė	point_n(geometry, index)
Argumentai	• **geometry** - geometry object • **index** - index of node to return, where 1 is the first node; if the value is negative, the selected vertex index will be its total count minus the absolute value
Pavyzdžiai	• `geom_to_wkt(point_n(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'),2))` → ‚Point (4 0)‘

Further reading: *Extract specific vertices* algorithm

point_on_surface

Returns a point guaranteed to lie on the surface of a geometry.

Sintaksė	point_on_surface(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `point_on_surface(@geometry)` → a point geometry

Further reading: *Point on Surface* algorithm

pole_of_inaccessibility

Calculates the approximate pole of inaccessibility for a surface, which is the most distant internal point from the boundary of the surface. This function uses the ‚polylabel' algorithm (Vladimir Agafonkin, 2016), which is an iterative approach guaranteed to find the true pole of inaccessibility within a specified tolerance. More precise tolerances require more iterations and will take longer to calculate.

Sintaksė	pole_of_inaccessibility(geometry, tolerance)
Argumentai	• **geometry** - a geometry • **tolerance** - maximum distance between the returned point and the true pole location
Pavyzdžiai	• `geom_to_wkt(pole_of_inaccessibility( geom_from_wkt('POLYGON((0 1, 0 9, 3 10, 3 3, 10 3, 10 1, 0 1))'), 0.1))` → ‚Point(1.546875 2.546875)'

9.26 Fig. : Pole of inaccessibility

Further reading: *Pole of inaccessibility* algorithm

project

Returns a point projected from a start point using a distance, a bearing (azimuth) and an elevation in radians.

Sintaksė	project(point, distance, azimuth, [elevation]) [] žymi neprivalomus argumentus
Argumentai	• **point** - start point • **distance** - distance to project • **azimuth** - azimuth in radians clockwise, where 0 corresponds to north • **elevation** - angle of inclination in radians
Pavyzdžiai	• `geom_to_wkt(project(make_point(1, 2), 3, radians(270)))` → ‚Point(-2, 2)'

Further reading: *Project points (Cartesian)* algorithm

relate

Tests the Dimensional Extended 9 Intersection Model (DE-9IM) representation of the relationship between two geometries.

Relationship variant

Returns the Dimensional Extended 9 Intersection Model (DE-9IM) representation of the relationship between two geometries.

Sintaksė	relate(geometry, geometry)
Argumentai	• **geometry** - a geometry • **geometry** - a geometry
Pavyzdžiai	• `relate( geom_from_wkt( 'LINESTRING(40 40,120 120)' ), geom_from_wkt( 'LINESTRING(40 40,60 120)' ) )` → ‚FF1F00102‘

Pattern match variant

Tests whether the DE-9IM relationship between two geometries matches a specified pattern.

Sintaksė	relate(geometry, geometry, pattern)
Argumentai	• **geometry** - a geometry • **geometry** - a geometry • **pattern** - DE-9IM pattern to match
Pavyzdžiai	• `relate( geom_from_wkt( 'LINESTRING(40 40,120 120)' ), geom_from_wkt( 'LINESTRING(40 40,60 120)' ), '**1F001**' )` → TRUE

reverse

Reverses the direction of a line string or reverses a string of text.

String variant

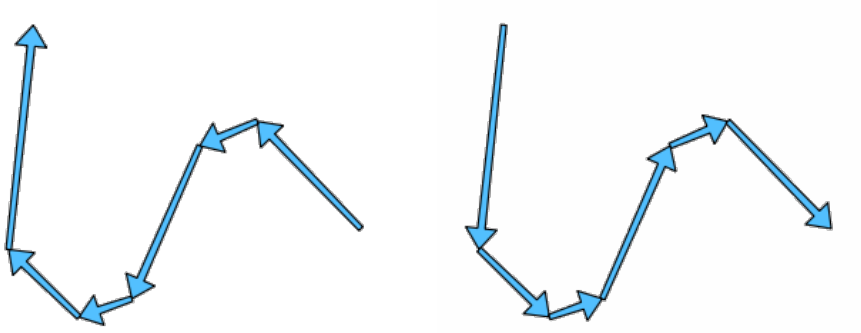
Reverses the order of characters in a string.

Sintaksė	reverse(string)
Argumentai	• **string** - string to reverse
Pavyzdžiai	• `reverse('hello')` → ‚olleh‘

Geometry variant

Reverses the direction of a line string by reversing the order of its vertices.

Sintaksė	reverse(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `geom_to_wkt(reverse(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → ‚LINESTRING(2 2, 1 1, 0 0)‘



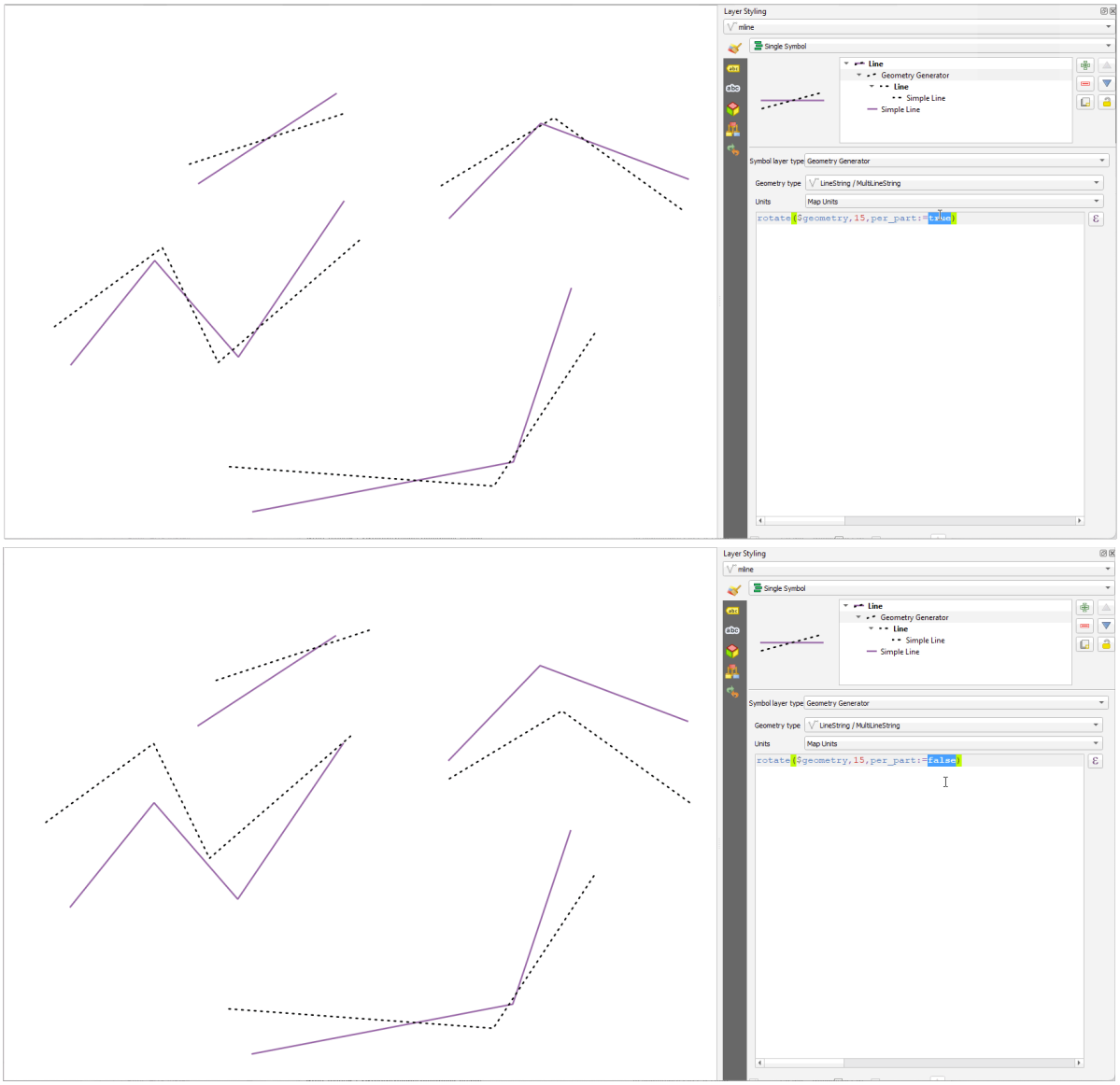
9.27 Fig. : Reversing line direction

Further reading: *Reverse line direction* algorithm

rotate

Returns a rotated version of a geometry. Calculations are in the Spatial Reference System of this geometry.

Sintaksė	rotate(geometry, rotation, [center=NULL], [per_part=false]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **rotation** - clockwise rotation in degrees • **center** - rotation center point. If not specified, the center of the geometry's bounding box is used. • **per_part** - apply rotation per part. If true, then rotation will apply around the center of each part's bounding box when the input geometry is multipart and an explicit rotation center point is not specified.
Pavyzdžiai	• `rotate(@geometry, 45, make_point(4, 5))` → geometry rotated 45 degrees clockwise around the (4, 5) point • `rotate(@geometry, 45)` → geometry rotated 45 degrees clockwise around the center of its bounding box



9.28 Fig. : Rotating features

roundness

Calculates how close a polygon shape is to a circle. The function returns 1 when the polygon shape is a perfect circle and 0 when it is completely flat.

Sintaksė	roundness(geometry)
Argumentai	• **geometry** - a polygon
Pavyzdžiai	• `round(roundness(geom_from_wkt('POLYGON(( 0 0, 0 1, 1 1, 1 0, 0 0))')), 3)` → 0.785 • `round(roundness(geom_from_wkt('POLYGON(( 0 0, 0 0.1, 1 0.1, 1 0, 0 0))')), 3)` → 0.260

Further reading: *Roundness* algorithm

scale

Returns a scaled version of a geometry. Calculations are in the Spatial Reference System of this geometry.

Sintaksė	scale(geometry, x_scale, y_scale, [center]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **x_scale** - x-axis scaling factor • **y_scale** - y-axis scaling factor • **center** - scaling center point. If not specified, the center of the geometry's bounding box is used.
Pavyzdžiai	• `scale(@geometry, 2, 0.5, make_point(4, 5))` → geometry scaled twice horizontally and halved vertically, around the (4, 5) point • `scale(@geometry, 2, 0.5)` → geometry twice horizontally and halved vertically, around the center of its bounding box

segments_to_lines

Returns a multi line geometry consisting of a line for every segment in the input geometry.

Sintaksė	segments_to_lines(geometry)
Argumentai	• **geometry** - geometry object
Pavyzdžiai	• `geom_to_wkt(segments_to_lines(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → ‚MultiLineString ((0 0, 1 1),(1 1, 2 2))‘

Further reading: *Explode lines* algorithm

shared_paths

Returns a collection containing paths shared by the two input geometries. Those going in the same direction are in the first element of the collection, those going in the opposite direction are in the second element. The paths themselves are given in the direction of the first geometry.

Sintaksė	shared_paths(geometry1, geometry2)
Argumentai	• **geometry1** - a LineString/MultiLineString geometry • **geometry2** - a LineString/MultiLineString geometry
Pavyzdžiai	• `geom_to_wkt(shared_paths(geom_from_wkt('MULTILINESTRING((26 125,26 200,126 200,126 125,26 125),(51 150,101 150,76 175,51 150))'),geom_from_wkt('LINESTRING(151 100,126 156.25,126 125,90 161, 76 175)'))` → ‚GeometryCollection (MultiLineString ((126 156.25, 126 125),(101 150, 90 161),(90 161, 76 175)),MultiLineString EMPTY)‘ • `geom_to_wkt(shared_paths(geom_from_wkt('LINESTRING(76 175,90 161,126 125,126 156.25,151 100)'), geom_from_wkt('MULTILINESTRING((26 125,26 200,126 200, 126 125,26 125),(51 150,101 150,76 175,51 150))'))` → ‚GeometryCollection (MultiLineString EMPTY,MultiLineString ((76 175, 90 161),(90 161, 101 150),(126 125, 126 156.25)))‘

shortest_line

Returns the shortest line joining geometry1 to geometry2. The resultant line will start at geometry1 and end at geometry2.

Sintaksė	shortest_line(geometry1, geometry2)
Argumentai	• **geometry1** - geometry to find shortest line from • **geometry2** - geometry to find shortest line to
Pavyzdžiai	• `geom_to_wkt(shortest_line(geom_from_wkt('LINESTRING (20 80, 98 190, 110 180, 50 75 )'),geom_from_wkt('POINT(100 100)')))` → ‚LineString(73.0769 115.384, 100 100)‘

simplify

Simplifies a geometry by removing nodes using a distance based threshold (ie, the Douglas Peucker algorithm). The algorithm preserves large deviations in geometries and reduces the number of vertices in nearly straight segments.

Sintaksė	simplify(geometry, tolerance)
Argumentai	• **geometry** - a geometry • **tolerance** - maximum deviation from straight segments for points to be removed
Pavyzdžiai	• `geom_to_wkt(simplify(geometry:=geom_from_wkt('LineString(0 0, 5 0.1, 10 0)'),tolerance:=5))` → ‚LineString(0 0, 10 0)‘

9.29 Fig. : From left to right, source layer and increasing simplification tolerances

Further reading: *Simplify* algorithm

simplify_vw

Simplifies a geometry by removing nodes using an area based threshold (ie, the Visvalingam-Whyatt algorithm). The algorithm removes vertices which create small areas in geometries, e.g., narrow spikes or nearly straight segments.

Sintaksė	simplify_vw(geometry, tolerance)
Argumentai	• **geometry** - a geometry • **tolerance** - a measure of the maximum area created by a node for the node to be removed
Pavyzdžiai	• `geom_to_wkt(simplify_vw(geometry:=geom_from_wkt('LineString(0 0, 5 0, 5.01 10, 5.02 0, 10 0)'),tolerance:=5))` → ‚LineString(0 0, 10 0)‘

Further reading: *Simplify* algorithm

single_sided_buffer

Returns a geometry formed by buffering out just one side of a linestring geometry. Distances are in the Spatial Reference System of this geometry.

Sintaksė	single_sided_buffer(geometry, distance, [segments=8], [join=1], [miter_limit=2.0]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a (multi)linestring geometry • **distance** - buffer distance. Positive values will be buffered to the left of lines, negative values to the right • **segments** - number of segments to use to represent a quarter circle when a round join style is used. A larger number results in a smoother buffer with more nodes. • **join** - join style for corners, where 1 = round, 2 = miter and 3 = bevel • **miter_limit** - limit on the miter ratio used for very sharp corners (when using miter joins only)
Pavyzdžiai	• `single_sided_buffer(@geometry, 10.5)` → line buffered to the left by 10.5 units • `single_sided_buffer(@geometry, -10.5)` → line buffered to the right by 10.5 units • `single_sided_buffer(@geometry, 10.5, segments:=16, join:=1)` → line buffered to the left by 10.5 units, using more segments to result in a smoother buffer • `single_sided_buffer(@geometry, 10.5, join:=3)` → line buffered to the left by 10.5 units, using a beveled join

9.30 Fig. : Left versus right side buffer on the same vector line layer

Further reading: *Single sided buffer* algorithm

sinuosity

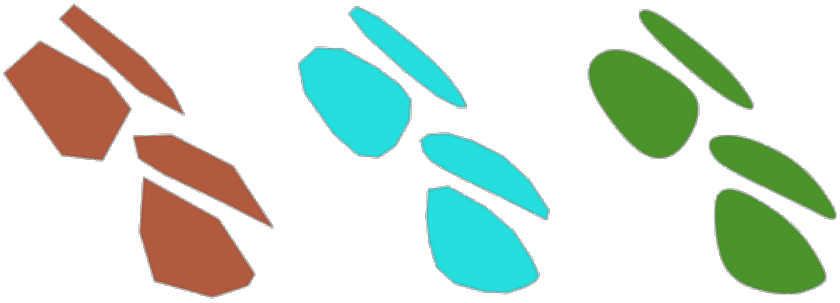
Returns the sinuosity of a curve, which is the ratio of the curve length to the straight (2D) distance between its endpoints.

Sintaksė	sinuosity(geometry)
Argumentai	• **geometry** - Input curve (circularstring, linestring)
Pavyzdžiai	• `round(sinuosity(geom_from_wkt('LINESTRING(2 0, 2 2, 3 2, 3 3)')), 3)` → 1.265 • `sinuosity(geom_from_wkt('LINESTRING( 3 1, 5 1)'))` → 1.0

smooth

Smooths a geometry by adding extra nodes which round off corners in the geometry. If input geometries contain Z or M values, these will also be smoothed and the output geometry will retain the same dimensionality as the input geometry.

Sintaksė	smooth(geometry, [iterations=1], [offset=0.25], [min_length=-1], [max_angle=180]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **iterations** - number of smoothing iterations to apply. Larger numbers result in smoother but more complex geometries. • **offset** - value between 0 and 0.5 which controls how tightly the smoothed geometry follow the original geometry. Smaller values result in a tighter smoothing, larger values result in looser smoothing. • **min_length** - minimum length of segments to apply smoothing to. This parameter can be used to avoid placing excessive additional nodes in shorter segments of the geometry. • **max_angle** - maximum angle at node for smoothing to be applied (0-180). By lowering the maximum angle intentionally sharp corners in the geometry can be preserved. For instance, a value of 80 degrees will retain right angles in the geometry.
Pavyzdžiai	• `geom_to_wkt(smooth(geometry:=geom_from_wkt('LineString(0 0, 5 0, 5 5)'),iterations:=1,offset:=0.2,min_length:=-1, max_angle:=180))` → ‚LineString (0 0, 4 0, 5 1, 5 5)‘



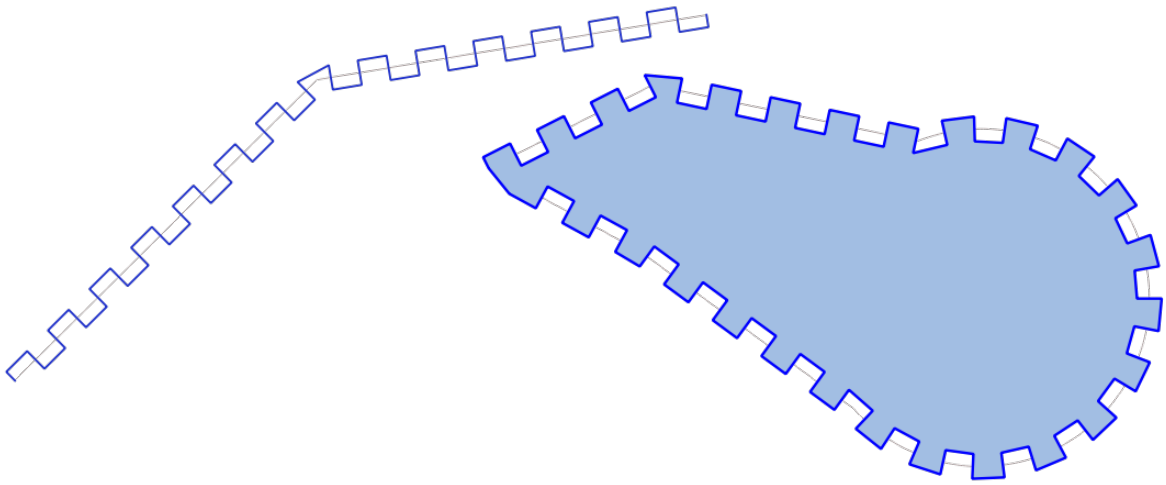
9.31 Fig. : Increasing number of iterations causes smoother geometries

Further reading: *Smooth* algorithm

square_wave

Constructs square/rectangular waves along the boundary of a geometry.

Sintaksė	square_wave(geometry, wavelength, amplitude, [strict=False]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **wavelength** - wavelength of square waveform • **amplitude** - amplitude of square waveform • **strict** - By default the wavelength argument is treated as a „maximum wavelength“, where the actual wavelength will be dynamically adjusted so that an exact number of square waves are created along the boundaries of the geometry. If the strict argument is set to true then the wavelength will be used exactly and an incomplete pattern may be used for the final waveform.
Pavyzdžiai	• `square_wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → Square waves with wavelength 3 and amplitude 1 along the linestring

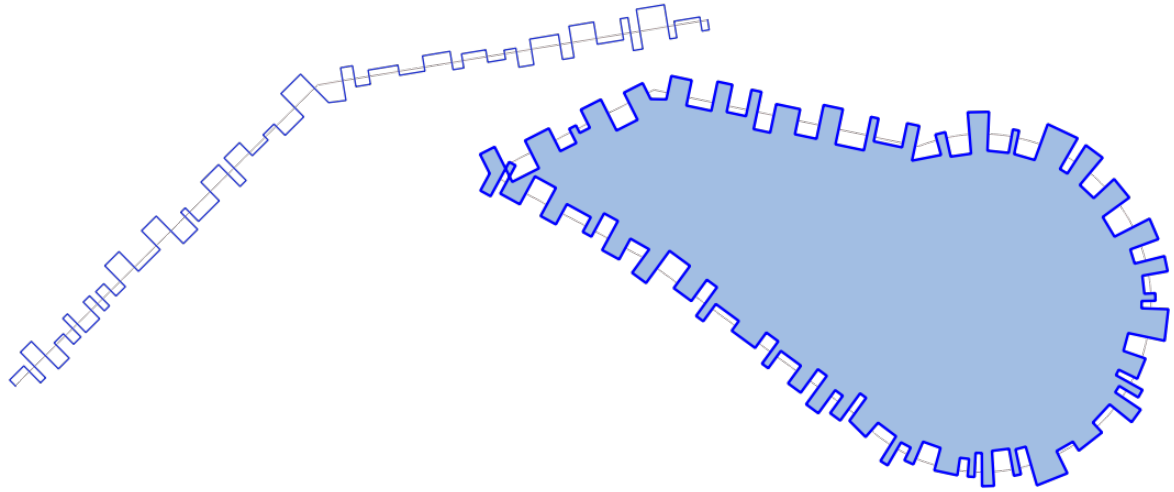


9.32 Fig. : Symbolizing features with square waves

square_wave_randomized

Constructs randomized square/rectangular waves along the boundary of a geometry.

Sintaksė	square_wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **min_wavelength** - minimum wavelength of waves • **max_wavelength** - maximum wavelength of waves • **min_amplitude** - minimum amplitude of waves • **max_amplitude** - maximum amplitude of waves • **seed** - specifies a random seed for generating waves. If the seed is 0, then a completely random set of waves will be generated.
Pavyzdžiai	• `square_wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → Randomly sized square waves with wavelengths between 2 and 3 and amplitudes between 0.1 and 0.2 along the linestring

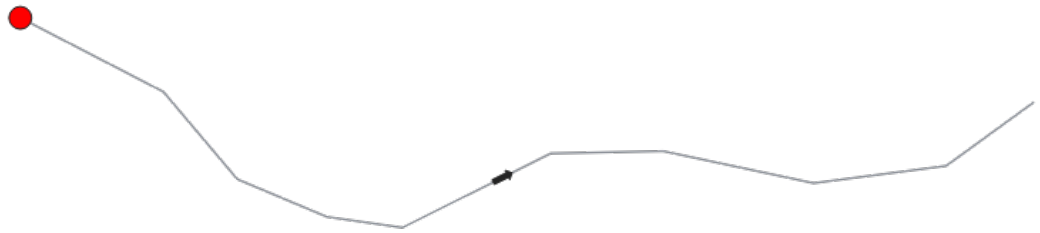


9.33 Fig. : Symbolizing features with square randomized waves

start_point

Returns the first node from a geometry.

Sintaksė	start_point(geometry)
Argumentai	• **geometry** - geometry object
Pavyzdžiai	• `geom_to_wkt(start_point(geom_from_wkt('LINESTRING(4 0, 4 2, 0 2)')))` → ‚Point (4 0)‘



9.34 Fig. : Starting point of a line feature

Further reading: *end_point*, *Extract specific vertices* algorithm

straight_distance_2d

Returns the direct/euclidean distance between the first and last vertex of a geometry. The geometry must be a curve (circularstring, linestring).

Sintaksė	straight_distance_2d(geometry)
Argumentai	• **geometry** - The geometry.
Pavyzdžiai	• `straight_distance_2d(geom_from_wkt('LINESTRING(1 0, 1 1)'))` → 1 • `round(straight_distance_2d(geom_from_wkt('LINESTRING(1 4, 3 5, 5 0)')), 3)` → 5.657

Further reading: *length*

sym_difference

Returns a geometry that represents the portions of two geometries that do not intersect.

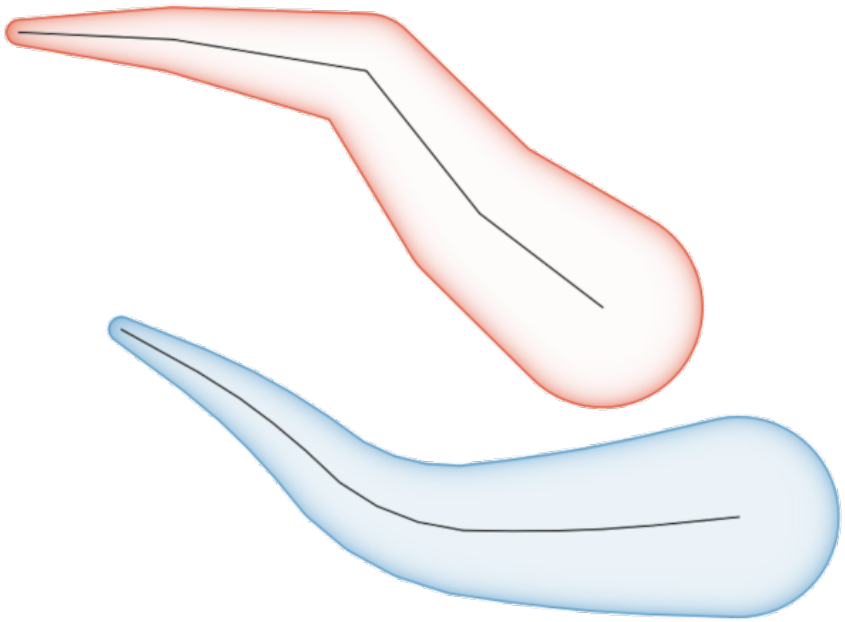
Sintaksė	sym_difference(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `geom_to_wkt( sym_difference( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 8 8)' ) ) )` → ‚LINESTRING(5 5, 8 8)‘

Further reading: *Symmetrical difference* algorithm

tapered_buffer

Creates a buffer along a line geometry where the buffer diameter varies evenly over the length of the line.

Sintaksė	tapered_buffer(geometry, start_width, end_width, [segments=8]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - input geometry. Must be a (multi)line geometry. • **start_width** - width of buffer at start of line, • **end_width** - width of buffer at end of line. • **segments** - number of segments to approximate quarter-circle curves in the buffer.
Pavyzdžiai	• `tapered_buffer(geometry:=geom_from_wkt('LINESTRING(1 2, 4 2)'),start_width:=1,end_width:=2,segments:=8)` → A tapered buffer starting with a diameter of 1 and ending with a diameter of 2 along the linestring geometry.



9.35 Fig. : Tapered buffer on line features

Further reading: *Tapered buffers* algorithm

touches

Tests whether a geometry touches another. Returns TRUE if the geometries have at least one point in common, but their interiors do not intersect.

Sintaksė	touches(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `touches( geom_from_wkt( 'LINESTRING(5 3, 4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `touches( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'POINT(5 5)' ) )` → FALSE

Further reading: *overlay_touches*

transform

Returns the geometry transformed from a source CRS to a destination CRS.

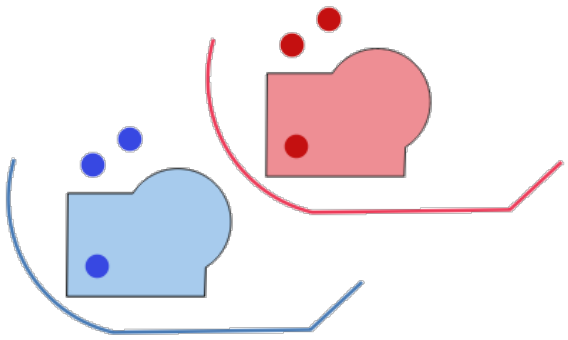
Sintaksė	transform(geometry, source_auth_id, dest_auth_id)
Argumentai	• **geometry** - a geometry • **source_auth_id** - the source CRS definition or CRS object • **dest_auth_id** - the destination CRS definition or CRS object
Pavyzdžiai	• `geom_to_wkt( transform( make_point(488995.53240249, 7104473.38600835), 'EPSG:2154', 'EPSG:4326' ) )` → ‚POINT(0 51)‘

Further reading: *Reproject layer* algorithm

translate

Returns a translated version of a geometry. Calculations are in the Spatial Reference System of this geometry.

Sintaksė	translate(geometry, dx, dy)
Argumentai	• **geometry** - a geometry • **dx** - delta x • **dy** - delta y
Pavyzdžiai	• `translate(@geometry, 5, 10)` → a geometry of the same type like the original one



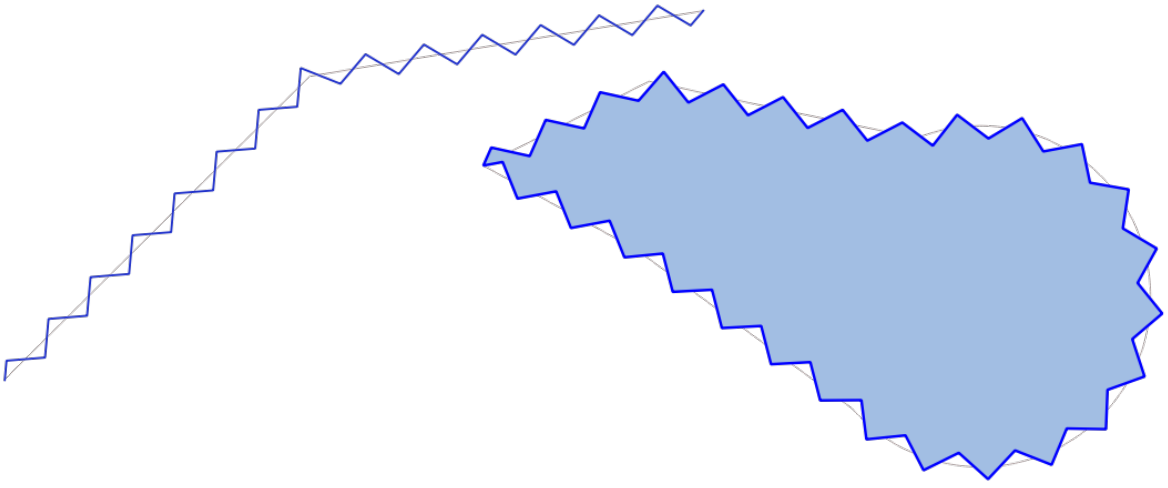
9.36 Fig. : Translating features

Further reading: *Translate* algorithm

triangular_wave

Constructs triangular waves along the boundary of a geometry.

Sintaksė	triangular_wave(geometry, wavelength, amplitude, [strict=False]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **wavelength** - wavelength of triangular waveform • **amplitude** - amplitude of triangular waveform • **strict** - By default the wavelength argument is treated as a „maximum wavelength“, where the actual wavelength will be dynamically adjusted so that an exact number of triangular waves are created along the boundaries of the geometry. If the strict argument is set to true then the wavelength will be used exactly and an incomplete pattern may be used for the final waveform.
Pavyzdžiai	• `triangular_wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → Triangular waves with wavelength 3 and amplitude 1 along the linestring

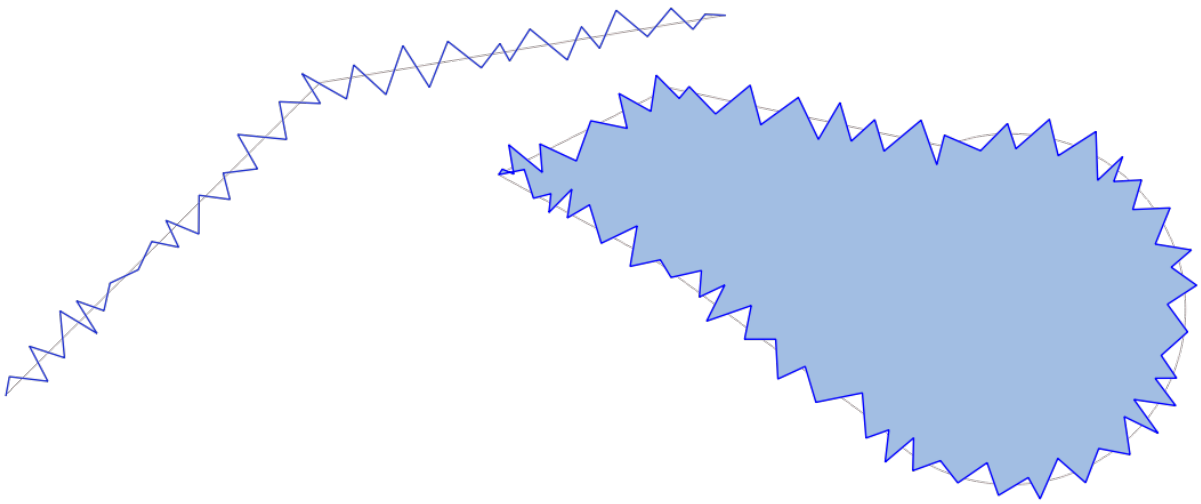


9.37 Fig. : Symbolizing features with triangular waves

triangular_wave_randomized

Constructs randomized triangular waves along the boundary of a geometry.

Sintaksė	triangular_wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **min_wavelength** - minimum wavelength of waves • **max_wavelength** - maximum wavelength of waves • **min_amplitude** - minimum amplitude of waves • **max_amplitude** - maximum amplitude of waves • **seed** - specifies a random seed for generating waves. If the seed is 0, then a completely random set of waves will be generated.
Pavyzdžiai	• `triangular_wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → Randomly sized triangular waves with wavelengths between 2 and 3 and amplitudes between 0.1 and 0.2 along the linestring



9.38 Fig. : Symbolizing features with triangular randomized waves

union

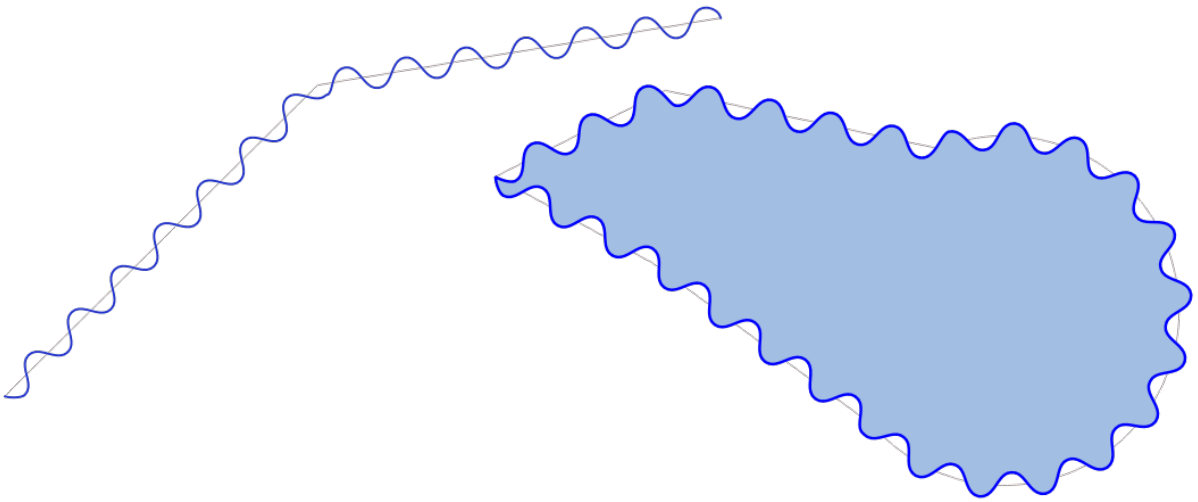
Returns a geometry that represents the point set union of the geometries.

Sintaksė	union(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `geom_to_wkt( union( make_point(4, 4), make_point(5, 5) ) )` → ‚MULTIPOINT(4 4, 5 5)‘

wave

Constructs rounded (sine-like) waves along the boundary of a geometry.

Sintaksė	wave(geometry, wavelength, amplitude, [strict=False]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **wavelength** - wavelength of sine-like waveform • **amplitude** - amplitude of sine-like waveform • **strict** - By default the wavelength argument is treated as a „maximum wavelength“, where the actual wavelength will be dynamically adjusted so that an exact number of waves are created along the boundaries of the geometry. If the strict argument is set to true then the wavelength will be used exactly and an incomplete pattern may be used for the final waveform.
Pavyzdžiai	• `wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → Sine-like waves with wavelength 3 and amplitude 1 along the linestring

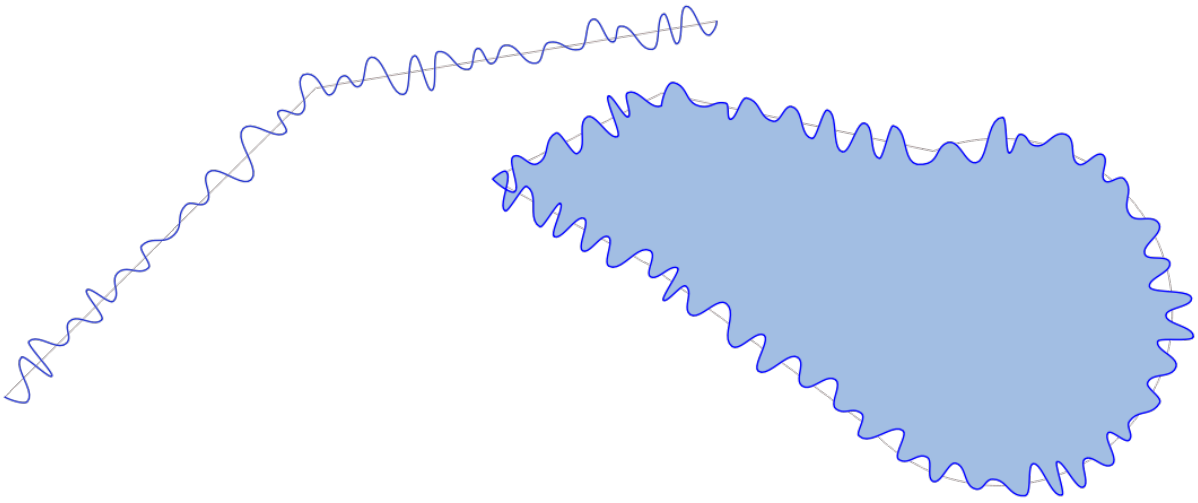


9.39 Fig. : Symbolizing features with waves

wave_randomized

Constructs randomized curved (sine-like) waves along the boundary of a geometry.

Sintaksė	wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0]) [] žymi neprivalomus argumentus
Argumentai	• **geometry** - a geometry • **min_wavelength** - minimum wavelength of waves • **max_wavelength** - maximum wavelength of waves • **min_amplitude** - minimum amplitude of waves • **max_amplitude** - maximum amplitude of waves • **seed** - specifies a random seed for generating waves. If the seed is 0, then a completely random set of waves will be generated.
Pavyzdžiai	• `wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → Randomly sized curved waves with wavelengths between 2 and 3 and amplitudes between 0.1 and 0.2 along the linestring

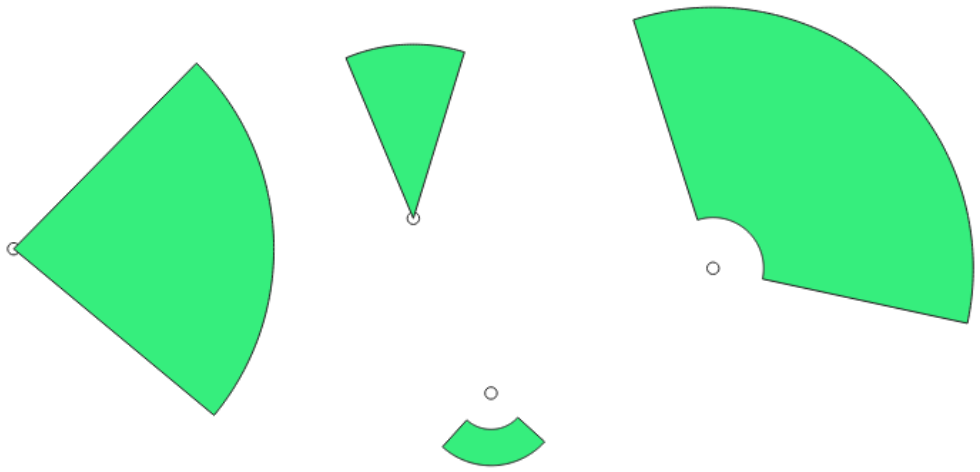


9.40 Fig. : Symbolizing features with randomized waves

wedge_buffer

Returns a wedge shaped buffer originating from a point geometry.

Sintaksė	wedge_buffer(center, azimuth, width, outer_radius, [inner_radius=0.0]) [] žymi neprivalomus argumentus
Argumentai	• **center** - center point (origin) of buffer. Must be a point geometry. • **azimuth** - angle (in degrees) for the middle of the wedge to point. • **width** - buffer width (in degrees). Note that the wedge will extend to half of the angular width either side of the azimuth direction. • **outer_radius** - outer radius for buffers • **inner_radius** - optional inner radius for buffers
Pavyzdžiai	• `wedge_buffer(center:=geom_from_wkt('POINT(1 2)'), azimuth:=90,width:=180,outer_radius:=1)` → A wedge shaped buffer centered on the point (1,2), facing to the East, with a width of 180 degrees and outer radius of 1.



9.41 Fig. : Wedge buffering features

Further reading: *Create wedge buffers* algorithm

within

Tests whether a geometry is within another. Returns TRUE if the geometry1 is completely within geometry2.

Sintaksė	within(geometry1, geometry2)
Argumentai	• **geometry1** - a geometry • **geometry2** - a geometry
Pavyzdžiai	• `within( geom_from_wkt( 'POINT( 0.5 0.5)' ), geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))' ) )` → TRUE • `within( geom_from_wkt( 'POINT( 5 5 )' ), geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0 ))' ) )` → FALSE

Further reading: *overlay_within*

$x

Returns the x coordinate of the current point feature. If the feature is a multipoint feature, then the x-coordinate of the first point will be returned. **WARNING: This function is deprecated. It is recommended to use the replacement x() function with @geometry variable instead.**

Sintaksė	$x
Pavyzdžiai	• `$x` → 42

Further reading: *x*

x

Returns the x coordinate of a point geometry, or the x coordinate of the centroid for a non-point geometry.

Sintaksė	x(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `x( geom_from_wkt( 'POINT(2 5)' ) )` → 2 • `x( @geometry )` → x coordinate of the current feature's centroid

$x_at

Retrieves a x coordinate of the current feature's geometry. **WARNING: This function is deprecated. It is recommended to use the replacement x_at function with @geometry variable instead.**

Sintaksė	$x_at(vertex)
Argumentai	• **vertex** - index of the vertex of the current geometry (indices start at 0; negative values apply from the last index, starting at -1)
Pavyzdžiai	• `$x_at(1)` → 5

Further reading: *x_at*

x_at

Retrieves a x coordinate of the geometry.

Sintaksė	x_at(geometry, vertex)
Argumentai	• **geometry** - geometry object • **vertex** - index of the vertex of the geometry (indices start at 0; negative values apply from the last index, starting at -1)
Pavyzdžiai	• `x_at( geom_from_wkt( 'POINT(4 5)' ), 0 )` → 4

x_max

Returns the maximum x coordinate of a geometry. Calculations are in the spatial reference system of this geometry.

Sintaksė	x_max(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `x_max( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 4

x_min

Returns the minimum x coordinate of a geometry. Calculations are in the spatial reference system of this geometry.

Sintaksė	x_min(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `x_min( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 2

$y

Returns the y coordinate of the current point feature. If the feature is a multipoint feature, then the y-coordinate of the first point will be returned. **WARNING: This function is deprecated. It is recommended to use the replacement y() function with @geometry variable instead.**

Sintaksė	$y
Pavyzdžiai	• `$y` → 42

Further reading: *y*

y

Returns the y coordinate of a point geometry, or the y coordinate of the centroid for a non-point geometry.

Sintaksė	y(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `y( geom_from_wkt( 'POINT(2 5)' ) )` → 5 • `y( @geometry )` → y coordinate of the current feature's centroid

$y_at

Retrieves a y coordinate of the current feature's geometry. **WARNING: This function is deprecated. It is recommended to use the replacement y_at function with @geometry variable instead.**

Sintaksė	$y_at(vertex)
Argumentai	• **vertex** - index of the vertex of the current geometry (indices start at 0; negative values apply from the last index, starting at -1)
Pavyzdžiai	• `$y_at(1)` → 2

Further reading: *y_at*

y_at

Retrieves a y coordinate of the geometry.

Sintaksė	y_at(geometry, vertex)
Argumentai	• **geometry** - geometry object • **vertex** - index of the vertex of the geometry (indices start at 0; negative values apply from the last index, starting at -1)
Pavyzdžiai	• `y_at( geom_from_wkt( 'POINT(4 5)' ), 0 )` → 5

y_max

Returns the maximum y coordinate of a geometry. Calculations are in the spatial reference system of this geometry.

Sintaksė	y_max(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `y_max( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 8

y_min

Returns the minimum y coordinate of a geometry. Calculations are in the spatial reference system of this geometry.

Sintaksė	y_min(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `y_min( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 5

$z

Returns the z value of the current point feature if it is 3D. If the feature is a multipoint feature, then the z value of the first point will be returned. **WARNING: This function is deprecated. It is recommended to use the replacement z() function with @geometry variable instead.**

Sintaksė	$z
Pavyzdžiai	• `$z` → 123

z

Returns the z coordinate of a point geometry, or NULL if the geometry has no z value.

Sintaksė	z(geometry)
Argumentai	• **geometry** - a point geometry
Pavyzdžiai	• `z( geom_from_wkt( 'POINTZ(2 5 7)' ) )` → 7

z_at

Retrieves a z coordinate of the geometry, or NULL if the geometry has no z value.

Sintaksė	z_at(geometry, vertex)
Argumentai	• **geometry** - geometry object • **vertex** - index of the vertex of the geometry (indices start at 0; negative values apply from the last index, starting at -1)
Pavyzdžiai	• `z_at(geom_from_wkt('LineStringZ(0 0 0, 10 10 5, 10 10 0)'), 1)` → 5

z_max

Returns the maximum z coordinate of a geometry, or NULL if the geometry has no z value.

Sintaksė	z_max(geometry)
Argumentai	• **geometry** - a geometry with z coordinate
Pavyzdžiai	• `z_max( geom_from_wkt( 'POINT ( 0 0 1 )' ) )` → 1 • `z_max( geom_from_wkt( 'MULTIPOINT ( 0 0 1 , 1 1 3 )' ) )` → 3 • `z_max( make_line( make_point( 0,0,0 ), make_point( -1,-1,-2 ) ) )` → 0 • `z_max( geom_from_wkt( 'LINESTRING( 0 0 0, 1 0 2, 1 1 -1 )' ) )` → 2 • `z_max( geom_from_wkt( 'POINT ( 0 0 )' ) )` → NULL

z_min

Returns the minimum z coordinate of a geometry, or NULL if the geometry has no z value.

Sintaksė	z_min(geometry)
Argumentai	• **geometry** - a geometry with z coordinate
Pavyzdžiai	• `z_min( geom_from_wkt( 'POINT ( 0 0 1 )' ) )` → 1 • `z_min( geom_from_wkt( 'MULTIPOINT ( 0 0 1 , 1 1 3 )' ) )` → 1 • `z_min( make_line( make_point( 0,0,0 ), make_point( -1,-1,-2 ) ) )` → -2 • `z_min( geom_from_wkt( 'LINESTRING( 0 0 0, 1 0 2, 1 1 -1 )' ) )` → -1 • `z_min( geom_from_wkt( 'POINT ( 0 0 )' ) )` → NULL

9.2.15 Layout Functions

This group contains functions to manipulate print layout items properties.

item_variables

Returns a map of variables from a layout item inside this print layout.

Sintaksė	item_variables(id)
Argumentai	• **id** - layout item ID
Pavyzdžiai	• `map_get( item_variables('Map 0'), 'map_scale')` → scale of the item ‚Map 0‘ in the current print layout

Further reading: List of default *variables*

map_credits

Returns a list of credit (usage rights) strings for the layers shown in a layout, or specific layout map item.

Sintaksė	map_credits([id], [include_layer_names=false], [layer_name_separator=': ']) [] žymi neprivalomus argumentus
Argumentai	• **id** - Map item ID. If not specified, the layers from all maps in the layout will be used. • **include_layer_names** - Set to true to include layer names before their credit strings • **layer_name_separator** - String to insert between layer names and their credit strings, if include_layer_names is true
Pavyzdžiai	• `array_to_string( map_credits() )` → comma separated list of layer credits for all layers shown in all map items in the layout, e.g 'CC-BY-NC, CC-BY-SA' • `array_to_string( map_credits( 'Main Map' ) )` → comma separated list of layer credits for layers shown in the 'Main Map' layout item, e.g 'CC-BY-NC, CC-BY-SA' • `array_to_string( map_credits( 'Main Map', include_layer_names := true, layer_name_separator := ': ' ) )` → comma separated list of layer names and their credits for layers shown in the 'Main Map' layout item, e.g. 'Railway lines: CC-BY-NC, Basemap: CC-BY-SA'

This function requires the *Access metadata properties* of the layers to have been filled.

9.2.16 Map Layers

This group contains a list of the available layers in the current project and, for each layer, their fields (stored in the dataset, virtual or auxiliary ones as well as from joins). The fields can be interacted the same way as mentioned in *Fields and Values*, except that a double-click will add the name as a string (single quoted) to the expression instead of as a field reference given that they do not belong to the active layer. This offers a convenient way to write expressions referring to different layers, such as when performing *aggregates*, *attribute* or *spatial* queries.

It also provides some convenient functions to manipulate layers.

decode_uri

Takes a layer and decodes the uri of the underlying data provider. It depends on the dataprovider, which data is available.

Sintaksė	decode_uri(layer, [part]) [] žymi neprivalomus argumentus
Argumentai	• **layer** - The layer for which the uri should be decoded. • **part** - The part of the uri to return. If unspecified, a map with all uri parts will be returned.
Pavyzdžiai	• `decode_uri(@layer)` → {'layerId': '0', 'layerName': '', 'path': '/home/qgis/shapefile.shp'} • `decode_uri(@layer)` → {'layerId': NULL, 'layerName': 'layer', 'path': '/home/qgis/geopackage.gpkg'} • `decode_uri(@layer, 'path')` → 'C:\my_data\qgis\shape.shp'

layer_property

Returns a matching layer property or metadata value.

Sintaksė	layer_property(layer, property)
Argumentai	• **layer** - tekstas, reprezentuojantis arba sluoksnio pavadinimą, arba sluoksnio ID • **property** - a string corresponding to the property to return. Valid options are: – name: layer name – id: layer ID – title: metadata title string – abstract: metadata abstract string – keywords: metadata keywords – data_url: metadata URL – attribution: metadata attribution string – attribution_url: metadata attribution URL – source: layer source – min_scale: minimum display scale for layer – max_scale: maximum display scale for layer – is_editable: if layer is in edit mode – crs: layer CRS – crs_definition: layer CRS full definition – crs_description: layer CRS description – crs_ellipsoid: acronym of the layer CRS ellipsoid – extent: layer extent (as a geometry object) – distance_units: layer distance units – type: layer type, e.g., Vector or Raster – storage_type: storage format (vector layers only) – geometry_type: geometry type, e.g., Point (vector layers only) – feature_count: approximate feature count for layer (vector layers only) – path: File path to the layer data source. Only available for file based layers.
Pavyzdžiai	• `layer_property('streets','title')` → ‚Basemap Streets‘ • `layer_property('airports','feature_count')` → 120 • `layer_property('landsat','crs')` → ‚EPSG:4326‘

Further reading: *vector*, *raster* and *mesh* layer properties

load_layer

Loads a layer by source URI and provider name.

Sintaksė	load_layer(uri, provider)
Argumentai	• **uri** - layer source URI string • **provider** - layer data provider name
Pavyzdžiai	• `layer_property(load_layer('c:/data/roads.shp', 'ogr'), 'feature_count')` → count of features from the c:/data/roads.shp vector layer

9.2.17 Maps Functions

This group contains functions to create or manipulate keys and values of map data structures (also known as dictionary objects, key-value pairs, or associative arrays). Unlike the *list data structure* where values order matters, the order of the key-value pairs in the map object is not relevant and values are identified by their keys.

from_json

Loads a JSON formatted string.

Sintaksė	from_json(string)
Argumentai	• **string** - JSON string
Pavyzdžiai	• `from_json('{"1":"one","2":"two"}')` → { ‚1‘: ‚one‘, ‚2‘: ‚two‘ } • `from_json('[1,2,3]')` → [1,2,3]

hstore_to_map

Creates a map from a hstore-formatted string.

Sintaksė	hstore_to_map(string)
Argumentai	• **string** - the input string
Pavyzdžiai	• `hstore_to_map('qgis=>rocks')` → { ‚qgis‘: ‚rocks‘ }

map

Returns a map containing all the keys and values passed as pair of parameters.

Sintaksė	map(key1, value1, key2, value2, …)
Argumentai	• **key** - a key (string) • **value** - a value
Pavyzdžiai	• `map('1','one','2', 'two')` → { ‚1‘: ‚one‘, ‚2‘: ‚two‘ } • `map('1','one','2', 'two')['1']` → ‚one‘

map_akeys

Returns all the keys of a map as an array.

Sintaksė	map_akeys(map)
Argumentai	• **map** - a map
Pavyzdžiai	• `map_akeys(map('1','one','2','two'))` → [‚1‘, ‚2‘]

map_avals

Returns all the values of a map as an array.

Sintaksė	map_avals(map)
Argumentai	• **map** - a map
Pavyzdžiai	• `map_avals(map('1','one','2','two'))` → [‚one‘, ‚two‘]

map_concat

Returns a map containing all the entries of the given maps. If two maps contain the same key, the value of the second map is taken.

Sintaksė	map_concat(map1, map2, …)
Argumentai	• **map** - a map
Pavyzdžiai	• `map_concat(map('1','one', '2','overridden'),map('2','two', '3','three'))` → { ‚1‘: ‚one‘, ‚2‘: ‚two‘, ‚3‘: ‚three‘ }

map_delete

Returns a map with the given key and its corresponding value deleted.

Sintaksė	map_delete(map, key)
Argumentai	• **map** - a map • **key** - the key to delete
Pavyzdžiai	• `map_delete(map('1','one','2','two'),'2')` → { ‚1‘: ‚one‘ }

map_exist

Returns TRUE if the given key exists in the map.

Sintaksė	map_exist(map, key)
Argumentai	• **map** - a map • **key** - the key to lookup
Pavyzdžiai	• `map_exist(map('1','one','2','two'),'3')` → FALSE

map_get

Returns the value of a map, given its key. Returns NULL if the key does not exist.

Sintaksė	map_get(map, key)
Argumentai	• **map** - a map • **key** - the key to lookup
Pavyzdžiai	• `map_get(map('1','one','2','two'),'2')` → ‚two‘ • `map_get( item_variables('Map 0'), 'map_scale')` → scale of the item ‚Map 0‘ (if it exists) in the current print layout

Patarimas

You can also use the *index operator ([])* to get a value from a map.

map_insert

Returns a map with an added key/value. If the key already exists, its value is overridden.

Sintaksė	map_insert(map, key, value)
Argumentai	• **map** - a map • **key** - the key to add • **value** - the value to add
Pavyzdžiai	• `map_insert(map('1','one'),'3','three')` → { ‚1‘: ‚one‘, ‚3‘: ‚three‘ } • `map_insert(map('1','one','2','overridden'),'2','two')` → { ‚1‘: ‚one‘, ‚2‘: ‚two‘ }

map_prefix_keys

Returns a map with all keys prefixed by a given string.

Sintaksė	map_prefix_keys(map, prefix)
Argumentai	• **map** - a map • **prefix** - a string
Pavyzdžiai	• `map_prefix_keys(map('1','one','2','two'), 'prefix-')` → { ‚prefix-1‘: ‚one‘, ‚prefix-2‘: ‚two‘ }

map_to_hstore

Merge map elements into a hstore-formatted string.

Sintaksė	map_to_hstore(map)
Argumentai	• **map** - the input map
Pavyzdžiai	• `map_to_hstore(map('1','one','2','two'))` → „„1“=>“one“‘„„2“=>“two“‘

map_to_html_dl

Merge map elements into a HTML definition list string.

Sintaksė	map_to_html_dl(map)
Argumentai	• **map** - the input map
Pavyzdžiai	• `map_to_html_dl(map('1','one','2','two'))` → <dl><dt>1</dt><dd>one</dd><dt>2</dt><dd>two</dd></dl>

map_to_html_table

Merge map elements into a HTML table string.

Sintaksė	map_to_html_table(map)
Argumentai	• **map** - the input map
Pavyzdžiai	• `map_to_html_table(map('1','one','2','two'))` → <table><thead><tr><th>1</th><th>2</th></tr></thead><tbody><tr><td>one</td><td>two</td></tr></tb

to_json

Create a JSON formatted string from a map, array or other value.

Sintaksė	to_json(value)
Argumentai	• **value** - The input value
Pavyzdžiai	• `to_json(map('1','one','2','two'))` → {„1“:“one“,“2“:“two“} • `to_json(array(1,2,3))` → [1,2,3]

url_encode

Returns an URL encoded string from a map. Transforms all characters in their properly-encoded form producing a fully-compliant query string.

Note that the plus sign ‚+‘ is not converted.

Sintaksė	url_encode(map)
Argumentai	• **map** - a map.
Pavyzdžiai	• `url_encode(map('a&+b', 'a and plus b', 'a=b', 'a equals b'))` → ‚a%26+b=a%20and%20plus%20b&a%3Db=a%20equals%20b‘

9.2.18 Mathematical Functions

This group contains math functions (e.g., square root, sin and cos).

abs

Returns the absolute value of a number.

Sintaksė	abs(value)
Argumentai	• **value** - a number
Pavyzdžiai	• `abs(-2)` → 2

acos

Returns the inverse cosine of a value in radians.

Sintaksė	acos(value)
Argumentai	• **value** - cosine of an angle in radians
Pavyzdžiai	• `acos(0.5)` → 1.0471975511966

asin

Returns the inverse sine of a value in radians.

Sintaksė	asin(value)
Argumentai	• **value** - sine of an angle in radians
Pavyzdžiai	• `asin(1.0)` → 1.5707963267949

atan

Returns the inverse tangent of a value in radians.

Sintaksė	atan(value)
Argumentai	• **value** - tan of an angle in radians
Pavyzdžiai	• `atan(0.5)` → 0.463647609000806

atan2

Returns the inverse tangent of dy/dx by using the signs of the two arguments to determine the quadrant of the result.

Sintaksė	atan2(dy, dx)
Argumentai	• **dy** - y coordinate difference • **dx** - x coordinate difference
Pavyzdžiai	• `atan2(1.0, 1.732)` → 0.523611477769969

ceil

Rounds a number upwards.

Sintaksė	ceil(value)
Argumentai	• **value** - a number
Pavyzdžiai	• `ceil(4.9)` → 5 • `ceil(-4.9)` → -4

clamp

Restricts an input value to a specified range.

Sintaksė	clamp(minimum, input, maximum)
Argumentai	• **minimum** - the smallest value *input* is allowed to take. • **input** - a value which will be restricted to the range specified by *minimum* and *maximum* • **maximum** - the largest value *input* is allowed to take
Pavyzdžiai	• `clamp(1,5,10)` → 5 *input* is between 1 and 10 so is returned unchanged • `clamp(1,0,10)` → 1 *input* is less than minimum value of 1, so function returns 1 • `clamp(1,11,10)` → 10 *input* is greater than maximum value of 10, so function returns 10

cos

Returns cosine of an angle.

Sintaksė	cos(angle)
Argumentai	• **angle** - angle in radians
Pavyzdžiai	• `cos(1.571)` → 0.000796326710733263

degrees

Converts from radians to degrees.

Sintaksė	degrees(radians)
Argumentai	• **radians** - numeric value
Pavyzdžiai	• `degrees(3.14159)` → 180 • `degrees(1)` → 57.2958

exp

Returns exponential of an value.

Sintaksė	exp(value)
Argumentai	• **value** - number to return exponent of
Pavyzdžiai	• `exp(1.0)` → 2.71828182845905

floor

Rounds a number downwards.

Sintaksė	floor(value)
Argumentai	• **value** - a number
Pavyzdžiai	• `floor(4.9)` → 4 • `floor(-4.9)` → -5

ln

Returns the natural logarithm of a value.

Sintaksė	ln(value)
Argumentai	• **value** - numeric value
Pavyzdžiai	• `ln(1)` → 0 • `ln(2.7182818284590452354)` → 1

log

Returns the value of the logarithm of the passed value and base.

Sintaksė	log(base, value)
Argumentai	• **base** - any positive number • **value** - any positive number
Pavyzdžiai	• `log(2, 32)` → 5 • `log(0.5, 32)` → -5

log10

Returns the value of the base 10 logarithm of the passed expression.

Sintaksė	log10(value)
Argumentai	• **value** - any positive number
Pavyzdžiai	• `log10(1)` → 0 • `log10(100)` → 2

max

Returns the largest value in a set of values.

Sintaksė	max(value1, value2, …)
Argumentai	• **value** - a number
Pavyzdžiai	• `max(2,10.2,5.5)` → 10.2 • `max(20.5,NULL,6.2)` → 20.5

min

Returns the smallest value in a set of values.

Sintaksė	min(value1, value2, …)
Argumentai	• **value** - a number
Pavyzdžiai	• `min(20.5,10,6.2)` → 6.2 • `min(2,-10.3,NULL)` → -10.3

pi

Returns value of pi for calculations.

Sintaksė	pi()
Pavyzdžiai	• `pi()` → 3.14159265358979

radians

Converts from degrees to radians.

Sintaksė	radians(degrees)
Argumentai	• **degrees** - numeric value
Pavyzdžiai	• `radians(180)` → 3.14159 • `radians(57.2958)` → 1

rand

Returns a random integer within the range specified by the minimum and maximum argument (inclusive). If a seed is provided, the returned will always be the same, depending on the seed.

Sintaksė	rand(min, max, [seed=NULL]) [] žymi neprivalomus argumentus
Argumentai	• **min** - an integer representing the smallest possible random number desired • **max** - an integer representing the largest possible random number desired • **seed** - any value to use as seed
Pavyzdžiai	• `rand(1, 10)` → 8

randf

Returns a random float within the range specified by the minimum and maximum argument (inclusive). If a seed is provided, the returned will always be the same, depending on the seed.

Sintaksė	randf([min=0.0], [max=1.0], [seed=NULL]) [] žymi neprivalomus argumentus
Argumentai	• **min** - an float representing the smallest possible random number desired • **max** - an float representing the largest possible random number desired • **seed** - any value to use as seed
Pavyzdžiai	• `randf(1, 10)` → 4.59258286403147

round

Rounds a number to number of decimal places.

Sintaksė	round(value, [places=0]) [] žymi neprivalomus argumentus
Argumentai	• **value** - decimal number to be rounded • **places** - Optional integer representing number of places to round decimals to. Can be negative.
Pavyzdžiai	• `round(1234.567, 2)` → 1234.57 • `round(1234.567)` → 1235 • `round(1234.567, -1)` → 1230

scale_exponential

Transforms a given value from an input domain to an output range using an exponential curve. This function can be used to ease values in or out of the specified output range.

Sintaksė	scale_exponential(value, domain_min, domain_max, range_min, range_max, exponent)
Argumentai	• **value** - A value in the input domain. The function will return a corresponding scaled value in the output range. • **domain_min** - Specifies the minimum value in the input domain, the smallest value the input value should take. • **domain_max** - Specifies the maximum value in the input domain, the largest value the input value should take. • **range_min** - Specifies the minimum value in the output range, the smallest value which should be output by the function. • **range_max** - Specifies the maximum value in the output range, the largest value which should be output by the function. • **exponent** - A positive value (greater than 0), which dictates the way input values are mapped to the output range. Large exponents will cause the output values to ‚ease in‘, starting slowly before accelerating as the input values approach the domain maximum. Smaller exponents (less than 1) will cause output values to ‚ease out‘, where the mapping starts quickly but slows as it approaches the domain maximum.
Pavyzdžiai	• `scale_exponential(5,0,10,0,100,2)` → 3.030 easing in, using an exponent of 2 • `scale_exponential(3,0,10,0,100,0.5)` → 87.585 easing out, using an exponent of 0.5

scale_linear

Transforms a given value from an input domain to an output range using linear interpolation.

Sintaksė	scale_linear(value, domain_min, domain_max, range_min, range_max)
Argumentai	• **value** - A value in the input domain. The function will return a corresponding scaled value in the output range. • **domain_min** - Specifies the minimum value in the input domain, the smallest value the input value should take. • **domain_max** - Specifies the maximum value in the input domain, the largest value the input value should take. • **range_min** - Specifies the minimum value in the output range, the smallest value which should be output by the function. • **range_max** - Specifies the maximum value in the output range, the largest value which should be output by the function.
Pavyzdžiai	• `scale_linear(5,0,10,0,100)` → 50 • `scale_linear(0.2,0,1,0,360)` → 72 scaling a value between 0 and 1 to an angle between 0 and 360 • `scale_linear(1500,1000,10000,9,20)` → 9.6111111 scaling a population which varies between 1000 and 10000 to a font size between 9 and 20

scale_polynomial

Transforms a given value from an input domain to an output range using a polynomial curve. This function can be used to ease values in or out of the specified output range.

Sintaksė	scale_polynomial(value, domain_min, domain_max, range_min, range_max, exponent)
Argumentai	• **value** - A value in the input domain. The function will return a corresponding scaled value in the output range. • **domain_min** - Specifies the minimum value in the input domain, the smallest value the input value should take. • **domain_max** - Specifies the maximum value in the input domain, the largest value the input value should take. • **range_min** - Specifies the minimum value in the output range, the smallest value which should be output by the function. • **range_max** - Specifies the maximum value in the output range, the largest value which should be output by the function. • **exponent** - A positive value (greater than 0), which dictates the way input values are mapped to the output range. Large exponents will cause the output values to ‚ease in', starting slowly before accelerating as the input values approach the domain maximum. Smaller exponents (less than 1) will cause output values to ‚ease out', where the mapping starts quickly but slows as it approaches the domain maximum.
Pavyzdžiai	• `scale_polynomial(5,0,10,0,100,2)` → 25 easing in, using an exponent of 2 • `scale_polynomial(3,0,10,0,100,0.5)` → 54.772 easing out, using an exponent of 0.5

sin

Returns the sine of an angle.

Sintaksė	sin(angle)
Argumentai	• **angle** - angle in radians
Pavyzdžiai	• `sin(1.571)` → 0.999999682931835

sqrt

Returns square root of a value.

Sintaksė	sqrt(value)
Argumentai	• **value** - a number
Pavyzdžiai	• `sqrt(9)` → 3

tan

Returns the tangent of an angle.

Sintaksė	tan(angle)
Argumentai	• **angle** - angle in radians
Pavyzdžiai	• `tan(1.0)` → 1.5574077246549

9.2.19 Meshes Functions

This group contains functions which calculate or return mesh related values.

$face_area

Returns the area of the current mesh face. The area calculated by this function respects both the current project's ellipsoid setting and area unit settings. For example, if an ellipsoid has been set for the project then the calculated area will be ellipsoidal, and if no ellipsoid is set then the calculated area will be planimetric.

Sintaksė	$face_area
Pavyzdžiai	• `$face_area` → 42

$face_index

Returns the index of the current mesh face.

Sintaksė	$face_index
Pavyzdžiai	• `$face_index` → 4581

$vertex_as_point

Returns the current vertex as a point geometry.

Sintaksė	$vertex_as_point
Pavyzdžiai	• `geom_to_wkt( $vertex_as_point )` → ‚POINT(800 1500 41)‘

$vertex_index

Returns the index of the current mesh vertex.

Sintaksė	$vertex_index
Pavyzdžiai	• `$vertex_index` → 9874

$vertex_x

Returns the X coordinate of the current mesh vertex.

Sintaksė	$vertex_x
Pavyzdžiai	• `$vertex_x` → 42.12

$vertex_y

Returns the Y coordinate of the current mesh vertex.

Sintaksė	$vertex_y
Pavyzdžiai	• `$vertex_y` → 12.24

$vertex_z

Returns the Z value of the current mesh vertex.

Sintaksė	$vertex_z
Pavyzdžiai	• `$vertex_z` → 42

9.2.20 Operators

This group contains operators (e.g., +, -, *). Note that for most of the mathematical functions below, if one of the inputs is NULL then the result is NULL.

%

Remainder of division. Takes the sign of the dividend.

Sintaksė	a % b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `9 % 2` → 1 • `9 % -2` → 1 • `-9 % 2` → -1 • `5 % NULL` → NULL

*

Multiplication of two values

Sintaksė	a * b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `5 * 4` → 20 • `5 * NULL` → NULL

+

Addition of two values. If one of the values is NULL the result will be NULL.

Sintaksė	a + b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `5 + 4` → 9 • `5 + NULL` → NULL • `'QGIS ' + 'ROCKS'` → ‚QGIS ROCKS‘ • `to_datetime('2020-08-01 12:00:00') + '1 day 2 hours'` → 2020-08-02T14:00:00

Further reading: *concat*, ||

-

Subtraction of two values. If one of the values is NULL the result will be NULL.

Sintaksė	a - b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `5 - 4` → 1 • `5 - NULL` → NULL • `to_datetime('2012-05-05 12:00:00') - to_interval('1 day 2 hours')` → 2012-05-04T10:00:00

/

Division of two values

Sintaksė	a / b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `5 / 4` → 1.25 • `5 / NULL` → NULL

//

Floor division of two values

Sintaksė	a // b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `9 // 2` → 4

<

Compares two values and evaluates to 1 if the left value is less than the right value.

Sintaksė	a < b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `5 < 4` → FALSE • `5 < 5` → FALSE • `4 < 5` → TRUE

<=

Compares two values and evaluates to 1 if the left value is less or equal than the right value.

Sintaksė	a <= b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `5 <= 4` → FALSE • `5 <= 5` → TRUE • `4 <= 5` → TRUE

<>

Compares two values and evaluates to 1 if they are not equal.

Sintaksė	a <> b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `5 <> 4` → TRUE • `4 <> 4` → FALSE • `5 <> NULL` → NULL • `NULL <> NULL` → NULL

=

Compares two values and evaluates to 1 if they are equal.

Sintaksė	a = b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `5 = 4` → FALSE • `4 = 4` → TRUE • `5 = NULL` → NULL • `NULL = NULL` → NULL

>

Compares two values and evaluates to 1 if the left value is greater than the right value.

Sintaksė	a > b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `5 > 4` → TRUE • `5 > 5` → FALSE • `4 > 5` → FALSE

>=

Compares two values and evaluates to 1 if the left value is greater or equal than the right value.

Sintaksė	a >= b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `5 >= 4` → TRUE • `5 >= 5` → TRUE • `4 >= 5` → FALSE

AND

Returns TRUE when conditions a and b are true.

Sintaksė	a AND b
Argumentai	• **a** - condition • **b** - condition
Pavyzdžiai	• `TRUE AND TRUE` → TRUE • `TRUE AND FALSE` → FALSE • `4 = 2+2 AND 1 = 1` → TRUE • `4 = 2+2 AND 1 = 2` → FALSE

BETWEEN

Returns TRUE if value is within the specified range. The range is considered inclusive of the bounds. To test for exclusion NOT BETWEEN can be used.

Sintaksė	value BETWEEN lower_bound AND higher_bound
Argumentai	• **value** - the value to compare with a range. It can be a string, a number or a date. • **lower_bound AND higher_bound** - range bounds
Pavyzdžiai	• `'B' BETWEEN 'A' AND 'C'` → TRUE • `2 BETWEEN 1 AND 3` → TRUE • `2 BETWEEN 2 AND 3` → TRUE • `'B' BETWEEN 'a' AND 'c'` → FALSE • `lower('B') BETWEEN 'a' AND 'b'` → TRUE

Pastaba

value BETWEEN lower_bound AND higher_bound is the same as „*value >= lower_bound AND value <= higher_bound*“.

Further reading: *NOT BETWEEN*

ILIKE

Returns TRUE if the first parameter matches case-insensitive the supplied pattern. LIKE can be used instead of ILIKE to make the match case-sensitive. Works with numbers also.

Sintaksė	string/number ILIKE pattern
Argumentai	• **string/number** - string to search • **pattern** - pattern to find, you can use ‚%‘ as a wildcard, ‚_‘ as a single char and ‚\\‘ to escape these special characters.
Pavyzdžiai	• `'A' ILIKE 'A'` → TRUE • `'A' ILIKE 'a'` → TRUE • `'A' ILIKE 'B'` → FALSE • `'ABC' ILIKE 'b'` → FALSE • `'ABC' ILIKE 'B'` → FALSE • `'ABC' ILIKE '_b_'` → TRUE • `'ABC' ILIKE '_B_'` → TRUE • `'ABCD' ILIKE '_b_'` → FALSE • `'ABCD' ILIKE '_B_'` → FALSE • `'ABCD' ILIKE '_b%'` → TRUE • `'ABCD' ILIKE '_B%'` → TRUE • `'ABCD' ILIKE '%b%'` → TRUE • `'ABCD' ILIKE '%B%'` → TRUE • `'ABCD%' ILIKE 'abcd\\%'` → TRUE • `'ABCD' ILIKE '%B\\%'` → FALSE

IN

Returns TRUE if value is found within a list of values.

Sintaksė	a IN b
Argumentai	• **a** - value • **b** - list of values
Pavyzdžiai	• `'A' IN ('A','B')` → TRUE • `'A' IN ('C','B')` → FALSE

IS

Returns TRUE if a is the same as b.

Sintaksė	a IS b
Argumentai	• **a** - any value • **b** - any value
Pavyzdžiai	• `'A' IS 'A'` → TRUE • `'A' IS 'a'` → FALSE • `4 IS 4` → TRUE • `4 IS 2+2` → TRUE • `4 IS 2` → FALSE • `@geometry IS NULL` → 0, if your geometry is not NULL

IS NOT

Returns TRUE if a is not the same as b.

Sintaksė	a IS NOT b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `'a' IS NOT 'b'` → TRUE • `'a' IS NOT 'a'` → FALSE • `4 IS NOT 2+2` → FALSE

LIKE

Returns TRUE if the first parameter matches the supplied pattern. Works with numbers also.

Sintaksė	string/number LIKE pattern
Argumentai	• **string/number** - value • **pattern** - pattern to compare value with, you can use ‚%‘ as a wildcard, ‚_‘ as a single char and ‚\\‘ to escape these special characters.
Pavyzdžiai	• `'A' LIKE 'A'` → TRUE • `'A' LIKE 'a'` → FALSE • `'A' LIKE 'B'` → FALSE • `'ABC' LIKE 'B'` → FALSE • `'ABC' LIKE '_B_'` → TRUE • `'ABCD' LIKE '_B_'` → FALSE • `'ABCD' LIKE '_B%'` → TRUE • `'ABCD' LIKE '%B%'` → TRUE • `'1%' LIKE '1\\%'` → TRUE • `'1_' LIKE '1\\%'` → FALSE

NOT

Negates a condition.

Sintaksė	NOT a
Argumentai	• **a** - condition
Pavyzdžiai	• `NOT 1` → FALSE • `NOT 0` → TRUE

NOT BETWEEN

Returns TRUE if value is not within the specified range. The range is considered inclusive of the bounds.

Sintaksė	value NOT BETWEEN lower_bound AND higher_bound
Argumentai	• **value** - the value to compare with a range. It can be a string, a number or a date. • **lower_bound AND higher_bound** - range bounds
Pavyzdžiai	• `'B' NOT BETWEEN 'A' AND 'C'` → FALSE • `1.0 NOT BETWEEN 1.1 AND 1.2` → TRUE • `2 NOT BETWEEN 2 AND 3` → FALSE • `'B' NOT BETWEEN 'a' AND 'c'` → TRUE • `lower('B') NOT BETWEEN 'a' AND 'b'` → FALSE

Pastaba

value NOT BETWEEN lower_bound AND higher_bound is the same as „*value < lower_bound OR value > higher_bound*“.

Further reading: *BETWEEN*

OR

Returns TRUE when condition a or b is true.

Sintaksė	a OR b
Argumentai	• **a** - condition • **b** - condition
Pavyzdžiai	• `4 = 2+2 OR 1 = 1` → TRUE • `4 = 2+2 OR 1 = 2` → TRUE • `4 = 2 OR 1 = 2` → FALSE

[]

Index operator. Returns an element from an array or map value.

Sintaksė	[index]
Argumentai	• **index** - array index or map key value
Pavyzdžiai	• `array(1,2,3)[0]` → 1 • `array(1,2,3)[2]` → 3 • `array(1,2,3)[-1]` → 3 • `map('a',1,'b',2)['a']` → 1 • `map('a',1,'b',2)['b']` → 2

Further reading: *array_get*, *map_get*

^

Power of two values.

Sintaksė	a ^ b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `5 ^ 4` → 625 • `5 ^ NULL` → NULL

||

Joins two values together into a string.

If one of the values is NULL the result will be NULL. See the CONCAT function for a different behavior.

Sintaksė	a || b
Argumentai	• **a** - value • **b** - value
Pavyzdžiai	• `'Here' || ' and ' || 'there'` → ‚Here and there‘ • `'Nothing' || NULL` → NULL • `'Dia: ' || "Diameter"` → ‚Dia: 25‘ • `1 || 2` → ‚12‘

Further reading: *concat*, *+*

~

Performs a regular expression match on a string value. Backslash characters must be double escaped (e.g., „\\s“ to match a white space character).

Sintaksė	string ~ regex
Argumentai	• **string** - A string value • **regex** - A regular expression. Slashes must be escaped, eg \\d.
Pavyzdžiai	• `'hello' ~ 'll'` → TRUE • `'hello' ~ '^ll'` → FALSE • `'hello' ~ 'llo$'` → TRUE • `'abc123' ~ '\\d+'` → TRUE

Further reading: *regexp_match*

9.2.21 Processing Functions

This group contains functions that operate on processing algorithms.

parameter

Returns the value of a processing algorithm input parameter.

Sintaksė	parameter(name)
Argumentai	• **name** - name of the corresponding input parameter
Pavyzdžiai	• `parameter('BUFFER_SIZE')` → 5.6

9.2.22 Rasters Functions

This group contains functions to operate on raster layer.

raster_attributes

Returns a map with the fields names as keys and the raster attribute table values as values from the attribute table entry that matches the given raster value.

Sintaksė	raster_attributes(layer, band, value)
Argumentai	• **layer** - the name or id of a raster layer • **band** - the band number for the associated attribute table lookup. • **value** - raster value
Pavyzdžiai	• `raster_attributes('vegetation', 1, raster_value('vegetation', 1, make_point(1,1)))` → {‚class‘: ‚Vegetated‘, ‚subclass‘: ‚Trees‘} • `raster_attributes('vegetation', 1, raster_value('vegetation', 1, @layer_cursor_point))` → {‚class‘: ‚Vegetated‘, ‚subclass‘: ‚Trees‘}

raster_statistic

Returns statistics from a raster layer.

Sintaksė	raster_statistic(layer, band, property)

Argumentai	• **layer** - a string, representing either a raster layer name or layer ID
	• **band** - integer representing the band number from the raster layer, starting at 1
	• **property** - a string corresponding to the property to return. Valid options are:
	  – min: minimum value
	  – max: maximum value
	  – avg: average (mean) value
	  – stdev: standard deviation of values
	  – range: range of values (max - min)
	  – sum: sum of all values from raster

Pavyzdžiai	• `raster_statistic('lc',1,'avg')` → Average value from band 1 from ‚lc‘ raster layer • `raster_statistic('ac2010',3,'min')` → Minimum value from band 3 from ‚ac2010‘ raster layer

raster_value

Returns the raster value found at the provided point.

Sintaksė	raster_value(layer, band, point)
Argumentai	• **layer** - the name or id of a raster layer • **band** - the band number to sample the value from. • **point** - point geometry (for multipart geometries having more than one part, a NULL value will be returned)
Pavyzdžiai	• `raster_value('dem', 1, make_point(1,1))` → 25 • `raster_value('ndvi', 2, @layer_cursor_point)` → 25

9.2.23 Record and Attributes Functions

This group contains functions that operate on record identifiers.

attribute

Returns an attribute from a feature.

Variant 1

Returns the value of an attribute from the current feature.

Sintaksė	attribute(attribute_name)
Argumentai	• **attribute_name** - name of attribute to be returned
Pavyzdžiai	• `attribute( 'name' )` → value stored in ‚name‘ attribute for the current feature

Variant 2

Allows the target feature and attribute name to be specified.

Sintaksė	attribute(feature, attribute_name)
Argumentai	• **feature** - a feature • **attribute_name** - name of attribute to be returned
Pavyzdžiai	• `attribute( @atlas_feature, 'name' )` → value stored in ‚name‘ attribute for the current atlas feature

attributes

Returns a map containing all attributes from a feature, with field names as map keys.

Variant 1

Returns a map of all attributes from the current feature.

Sintaksė	attributes()
Pavyzdžiai	• `attributes()['name']` → value stored in ‚name‘ attribute for the current feature

Variant 2

Allows the target feature to be specified.

Sintaksė	attributes(feature)
Argumentai	• **feature** - a feature
Pavyzdžiai	• `attributes( @atlas_feature )['name']` → value stored in ‚name‘ attribute for the current atlas feature

Further reading: *Maps Functions*

$currentfeature

Returns the current feature being evaluated. This can be used with the ‚attribute' function to evaluate attribute values from the current feature. **WARNING: This function is deprecated. It is recommended to use the replacement @feature variable instead.**

Sintaksė	$currentfeature
Pavyzdžiai	• `attribute( $currentfeature, 'name' )` → value stored in ‚name' attribute for the current feature

display_expression

Returns the display expression for a given feature in a layer. The expression is evaluated by default. Can be used with zero, one or more arguments, see below for details.

No parameters

If called with no parameters, the function will evaluate the display expression of the current feature in the current layer.

Sintaksė	display_expression()
Pavyzdžiai	• `display_expression()` → The display expression of the current feature in the current layer.

One ‚feature' parameter

If called with a ‚feature' parameter only, the function will evaluate the specified feature from the current layer.

Sintaksė	display_expression(feature)
Argumentai	• **feature** - The feature which should be evaluated.
Pavyzdžiai	• `display_expression(@atlas_feature)` → The display expression of the current atlas feature.

Layer and feature parameters

If the function is called with both a layer and a feature, it will evaluate the specified feature from the specified layer.

Sintaksė	display_expression(layer, feature, [evaluate=true]) [] žymi neprivalomus argumentus
Argumentai	• **layer** - The layer (or its ID or name) • **feature** - The feature which should be evaluated. • **evaluate** - If the expression must be evaluated. If false, the expression will be returned as a string literal only (which could potentially be later evaluated using the ‚eval' function).
Pavyzdžiai	• `display_expression( 'streets', get_feature_by_id('streets', 1))` → The display expression of the feature with the ID 1 on the layer ‚streets'. • `display_expression('a_layer_id', @feature, 'False')` → The display expression of the given feature not evaluated.

feature_id

Returns a feature's unique ID, or NULL if the feature is not valid.

Sintaksė	feature_id(feature)
Argumentai	• **feature** - a feature object
Pavyzdžiai	• `feature_id( @feature )` → the ID of the current feature

Further reading: *get_feature_by_id*

get_feature

Returns the first feature of a layer matching a given attribute value.

Single value variant

Along with the layer ID, a single column and value are specified.

Sintaksė	get_feature(layer, attribute, value)
Argumentai	• **layer** - layer name or ID • **attribute** - attribute name to use for the match • **value** - attribute value to match
Pavyzdžiai	• `get_feature('streets','name','main st')` → first feature found in „streets" layer with „main st" value in the „name" field

Map variant

Along with the layer ID, a map containing the columns (key) and their respective value to be used.

Sintaksė	get_feature(layer, attribute)
Argumentai	• **layer** - layer name or ID • **attribute** - Map containing the column and value pairs to use
Pavyzdžiai	• `get_feature('streets',map('name','main st','lane_num','4'))` → first feature found in „streets“ layer with „main st“ value in the „name“ field and „4“ value in the „lane_num“ field

get_feature_by_id

Returns the feature with an id on a layer.

Sintaksė	get_feature_by_id(layer, feature_id)
Argumentai	• **layer** - layer, layer name or layer id • **feature_id** - the id of the feature which should be returned
Pavyzdžiai	• `get_feature_by_id('streets', 1)` → the feature with the id 1 on the layer „streets“

Further reading: *feature_id*

$id

Returns the feature id of the current row. **WARNING: This function is deprecated. It is recommended to use the replacement @id variable instead.**

Sintaksė	$id
Pavyzdžiai	• `$id` → 42

Further reading: *feature_id*, *get_feature_by_id*

is_attribute_valid

Returns TRUE if a specific feature attribute meets all constraints.

Sintaksė	is_attribute_valid(attribute, [feature], [layer], [strength]) [] žymi neprivalomus argumentus
Argumentai	• **attribute** - an attribute name • **feature** - A feature. If not set, the current feature will be used. • **layer** - A vector layer. If not set, the current layer will be used. • **strength** - Set to ‚hard' or ‚soft' to narrow down to a specific constraint type. If not set, the function will return FALSE if either a hard or a soft constraint fails.
Pavyzdžiai	• `is_attribute_valid('HECTARES')` → TRUE if the current feature's value in the „HECTARES" field meets all constraints. • `is_attribute_valid('HOUSES',get_feature('my_layer', 'FID', 10), 'my_layer')` → FALSE if the value in the „HOUSES" field from the feature with „FID"=10 in ‚my_layer' fails to meet all constraints.

Further reading: *Constraints*

is_feature_valid

Returns TRUE if a feature meets all field constraints.

Sintaksė	is_feature_valid([feature], [layer], [strength]) [] žymi neprivalomus argumentus
Argumentai	• **feature** - A feature. If not set, the current feature will be used. • **layer** - A vector layer. If not set, the current layer will be used. • **strength** - Set to ‚hard' or ‚soft' to narrow down to a specific constraint type. If not set, the function will return FALSE if either a hard or a soft constraint fails.
Pavyzdžiai	• `is_feature_valid(strength:='hard')` → TRUE if all fields from the current feature meet their hard constraints. • `is_feature_valid(get_feature('my_layer', 'FID', 10), 'my_layer')` → FALSE if all fields from feature with „FID"=10 in ‚my_layer' fails to meet all constraints.

Further reading: *Constraints*

is_selected

Returns TRUE if a feature is selected. Can be used with zero, one or two arguments, see below for details.

No parameters

If called with no parameters, the function will return TRUE if the current feature in the current layer is selected.

Sintaksė	is_selected()
Pavyzdžiai	• `is_selected()` → TRUE if the current feature in the current layer is selected.

One ‚feature' parameter

If called with a ‚feature' parameter only, the function returns TRUE if the specified feature from the current layer is selected.

Sintaksė	is_selected(feature)
Argumentai	• **feature** - The feature which should be checked for selection.
Pavyzdžiai	• `is_selected(@atlas_feature)` → TRUE if the current atlas feature is selected. • `is_selected(get_feature('streets', 'name', 'Main St.'))` → TRUE if the unique named „Main St." feature on the active „streets" layer is selected. • `is_selected(get_feature_by_id('streets', 1))` → TRUE if the feature with the id 1 on the active „streets" layer is selected.

Two parameters

If the function is called with both a layer and a feature, it will return TRUE if the specified feature from the specified layer is selected.

Sintaksė	is_selected(layer, feature)
Argumentai	• **layer** - The layer (its ID or name) on which the selection will be checked. • **feature** - The feature which should be checked for selection.
Pavyzdžiai	• `is_selected( 'streets', get_feature('streets', 'name', "street_name"))` → TRUE if the current building's street is selected (assuming the building layer has a field named ‚street_name' and the ‚streets' layer has a field called ‚name' with unique values). • `is_selected( 'streets', get_feature_by_id('streets', 1))` → TRUE if the feature with the id 1 on the „streets" layer is selected.

maptip

Returns the maptip for a given feature in a layer. The expression is evaluated by default. Can be used with zero, one or more arguments, see below for details.

No parameters

If called with no parameters, the function will evaluate the maptip of the current feature in the current layer.

Sintaksė	maptip()
Pavyzdžiai	• `maptip()` → The maptip of the current feature in the current layer.

One ‚feature' parameter

If called with a ‚feature' parameter only, the function will evaluate the specified feature from the current layer.

Sintaksė	maptip(feature)
Argumentai	• **feature** - The feature which should be evaluated.
Pavyzdžiai	• `maptip(@atlas_feature)` → The maptip of the current atlas feature.

Layer and feature parameters

If the function is called with both a layer and a feature, it will evaluate the specified feature from the specified layer.

Sintaksė	maptip(layer, feature, [evaluate=true]) [] žymi neprivalomus argumentus
Argumentai	• **layer** - The layer (or its ID or name) • **feature** - The feature which should be evaluated. • **evaluate** - If the expression must be evaluated. If false, the expression will be returned as a string literal only (which could potentially be later evaluated using the ‚eval_template' function).
Pavyzdžiai	• `maptip('streets', get_feature_by_id('streets', 1))` → The maptip of the feature with the ID 1 on the layer ‚streets'. • `maptip('a_layer_id', @feature, 'False')` → The maptip of the given feature not evaluated.

num_selected

Returns the number of selected features on a given layer. By default works on the layer on which the expression is evaluated.

Sintaksė	num_selected([layer=current layer]) [] žymi neprivalomus argumentus
Argumentai	• **layer** - The layer (or its id or name) on which the selection will be checked.
Pavyzdžiai	• `num_selected()` → The number of selected features on the current layer. • `num_selected('streets')` → The number of selected features on the layer streets

represent_attributes

Returns a map with the attribute names as keys and the configured representation values as values. The representation value for the attributes depends on the configured widget type for each attribute. Can be used with zero, one or more arguments, see below for details.

No parameters

If called with no parameters, the function will return the representation of the attributes of the current feature in the current layer.

Sintaksė	represent_attributes()
Pavyzdžiai	• `represent_attributes()` → The representation of the attributes for the current feature.

One ‚feature' parameter

If called with a ‚feature' parameter only, the function will return the representation of the attributes of the specified feature from the current layer.

Sintaksė	represent_attributes(feature)
Argumentai	• **feature** - The feature which should be evaluated.
Pavyzdžiai	• `represent_attributes(@atlas_feature)` → The representation of the attributes for the specified feature from the current layer.

Layer and feature parameters

If called with a ‚layer' and a ‚feature' parameter, the function will return the representation of the attributes of the specified feature from the specified layer.

Sintaksė	represent_attributes(layer, feature)
Argumentai	• **layer** - The layer (or its ID or name). • **feature** - The feature which should be evaluated.
Pavyzdžiai	• `represent_attributes('atlas_layer', @atlas_feature)` → The representation of the attributes for the specified feature from the specified layer.

Further reading: *represent_value*

represent_value

Returns the configured representation value for a field value. It depends on the configured widget type. Often, this is useful for ‚Value Map' widgets.

Sintaksė	represent_value(value, [fieldName]) [] žymi neprivalomus argumentus
Argumentai	• **value** - The value which should be resolved. Most likely a field. • **fieldName** - The field name for which the widget configuration should be loaded.
Pavyzdžiai	• `represent_value("field_with_value_map")` → Description for value • `represent_value('static value', 'field_name')` → Description for static value

Further reading: *widget types*, *represent_attributes*

sqlite_fetch_and_increment

Manage autoincrementing values in sqlite databases.

SQlite default values can only be applied on insert and not prefetched.

This makes it impossible to acquire an incremented primary key via AUTO_INCREMENT before creating the row in the database. Sidenote: with postgres, this works via the option *evaluate default values*.

When adding new features with relations, it is really nice to be able to already add children for a parent, while the parents form is still open and hence the parent feature uncommitted.

To get around this limitation, this function can be used to manage sequence values in a separate table on sqlite based formats like gpkg.

The sequence table will be filtered for a sequence id (filter_attribute and filter_value) and the current value of the id_field will be incremented by 1 and the incremented value returned.

If additional columns require values to be specified, the default_values map can be used for this purpose.

Note

This function modifies the target sqlite table. It is intended for usage with default value configurations for attributes.

When the database parameter is a layer and the layer is in transaction mode, the value will only be retrieved once during the lifetime of a transaction and cached and incremented. This makes it unsafe to work on the same database from several processes in parallel.

Sintaksė	sqlite_fetch_and_increment(database, table, id_field, filter_attribute, filter_value, [default_values]) [] žymi neprivalomus argumentus
Argumentai	• **database** - Path to the sqlite file or geopackage layer • **table** - Name of the table that manages the sequences • **id_field** - Name of the field that contains the current value • **filter_attribute** - Name the field that contains a unique identifier for this sequence. Must have a UNIQUE index. • **filter_value** - Name of the sequence to use. • **default_values** - Map with default values for additional columns on the table. The values need to be fully quoted. Functions are allowed.
Pavyzdžiai	• `sqlite_fetch_and_increment(@layer, 'sequence_table', 'last_unique_id', 'sequence_id', 'global', map('last_change', 'date(''now'')', 'user', '''' || @user_account_name || '''')) → 0` • `sqlite_fetch_and_increment(layer_property(@layer, 'path'), 'sequence_table', 'last_unique_id', 'sequence_id', 'global', map('last_change', 'date(''now'')', 'user', '''' || @user_account_name || '''')) → 0`

Further reading: *Duomenų šaltinių savybės*, *Setting relations between multiple layers*

uuid

Generates a Universally Unique Identifier (UUID) for each row using the Qt QUuid::createUuid method.

Sintaksė	uuid([format='WithBraces']) [] žymi neprivalomus argumentus
Argumentai	• **format** - The format, as the UUID will be formatted. ‚WithBraces', ‚WithoutBraces' or ‚Id128'.
Pavyzdžiai	• `uuid()` → ‚{0bd2f60f-f157-4a6d-96af-d4ba4cb366a1}' • `uuid('WithoutBraces')` → ‚0bd2f60f-f157-4a6d-96af-d4ba4cb366a1' • `uuid('Id128')` → ‚0bd2f60ff1574a6d96afd4ba4cb366a1'

9.2.24 Relations

This group contains the list of the *relations* available in the current project, with their description. It provides a quick access to the relation ID for writing an expression (with e.g. the *relation_aggregate* function) or customizing a form.

9.2.25 Sensors Functions

This group contains functions to interact with sensors.

sensor_data

Returns the last captured value (or values as a map for sensors which report multiple values) from a registered sensor.

Sintaksė	sensor_data(name, [expiration]) [] žymi neprivalomus argumentus
Argumentai	• **name** - the sensor name • **expiration** - maximum millisecond since last captured value allowed
Pavyzdžiai	• `sensor_data('geiger_1')` → ‚2000‘

9.2.26 String Functions

This group contains functions that operate on strings (e.g., that replace, convert to upper case).

ascii

Returns the unicode code associated with the first character of a string.

Sintaksė	ascii(string)
Argumentai	• **string** - the string to convert to unicode code
Pavyzdžiai	• `ascii('Q')` → 81

char

Returns the character associated with a unicode code.

Sintaksė	char(code)
Argumentai	• **code** - a unicode code number
Pavyzdžiai	• `char(81)` → ‚Q‘

concat

Concatenates several strings to one. NULL values are converted to empty strings. Other values (like numbers) are converted to strings.

Sintaksė	concat(string1, string2, …)
Argumentai	• **string** - a string value
Pavyzdžiai	• `concat('sun', 'set')` → ‚sunset‘ • `concat('a','b','c','d','e')` → ‚abcde‘ • `concat('Anno ', 1984)` → ‚Anno 1984‘ • `concat('The Wall', NULL)` → ‚The Wall‘

About fields concatenation

You can also concatenate strings or field values using either `||` or + operators, with some special characteristics:

- The + operator also means sum up expression, so if you have an integer (field or numeric value) operand, this can be error prone and you better use the others:

```
'My feature id is: ' + "gid" => triggers an error as gid returns an integer
```

- When any of the arguments is a NULL value, either `||` or + will return a NULL value. To return the other arguments regardless the NULL value, you may want to use the `concat` or `array_to_string` function:

```
'My feature id is: ' + NULL ==> NULL
'My feature id is: ' || NULL => NULL
concat('My feature id is: ', NULL) => 'My feature id is: '
array_to_string( array('My feature id is: ', NULL) ) => 'My feature id is: '
```

further reading: *array_to_string*, ||, +

format

Format a string using supplied arguments.

Sintaksė	format(string, arg1, arg2, …)
Argumentai	• **string** - A string with placeholders %1, %2, etc., for the arguments. Placeholders can be repeated. The lowest numbered placeholder is replaced by arg1, the next by arg2, etc. • **arg** - any type. Any number of arguments.
Pavyzdžiai	• `format('This %1 a %2','is', 'test')` → ‚This is a test‘ • `format('This is %2','a bit unexpected but 2 is lowest number in string','normal')` → ‚This is a bit unexpected but 2 is lowest number in string‘

format_date

Formats a date type or string into a custom string format. Uses Qt date/time format strings. See QDateTime::toString.

Sintaksė	format_date(datetime, format, [language]) [] žymi neprivalomus argumentus
Argumentai	• **datetime** - date, time or datetime value • **format** - String template used to format the string.

Expression	Output
d	the day as number without a leading zero (1 to 31)
dd	the day as number with a leading zero (01 to 31)
ddd	the abbreviated localized day name (e.g. ‚Mon' to ‚Sun')
dddd	the long localized day name (e.g. ‚Monday' to ‚Sunday')
M	the month as number without a leading zero (1-12)
MM	the month as number with a leading zero (01-12)
MMM	the abbreviated localized month name (e.g. ‚Jan' to ‚Dec')
MMMM	the long localized month name (e.g. ‚January' to ‚December')
yy	the year as two digit number (00-99)
yyyy	the year as four digit number

These expressions may be used for the time part of the format string:

Expression	Output
h	the hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)
hh	the hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)
H	the hour without a leading zero (0 to 23, even with AM/PM display)
HH	the hour with a leading zero (00 to 23, even with AM/PM display)
m	the minute without a leading zero (0 to 59)
mm	the minute with a leading zero (00 to 59)
s	the second without a leading zero (0 to 59)
ss	the second with a leading zero (00 to 59)
z	the milliseconds without trailing zeroes (0 to 999)
zzz	the milliseconds with trailing zeroes (000 to 999)
AP or A	interpret as an AM/PM time. *AP* must be either ‚AM' or ‚PM'.
ap or a	Interpret as an AM/PM time. *ap* must be either ‚am' or ‚pm'.

- **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to format the date into a custom string. By default the current QGIS user locale is used.

Pavyzdžiai

- `format_date('2012-05-15','dd.MM.yyyy')` → ‚15.05.2012'
- `format_date('2012-05-15','d MMMM yyyy','fr')` → ‚15 mai 2012'
- `format_date('2012-05-15','dddd')` → ‚Tuesday', if the current locale is an English variant
- `format_date('2012-05-15 13:54:20','dd.MM.yy')` → ‚15.05.12'
- `format_date('13:54:20','hh:mm AP')` → ‚01:54 PM'

format_number

Returns a number formatted with the locale separator for thousands. By default the current QGIS user locale is used. Also truncates the decimal places to the number of supplied places.

Sintaksė	format_number(number, [places=0], [language], [omit_group_separators=false], [trim_trailing_zeroes=false]) [] žymi neprivalomus argumentus
Argumentai	• **number** - number to be formatted • **places** - integer representing the number of decimal places to truncate the string to. • **language** - language (lowercase, two- or three-letter, ISO 639 language code) used to format the number into a string. By default the current QGIS user locale is used. • **omit_group_separators** - if set to true then group separators will not be included in the string • **trim_trailing_zeroes** - if set to true then trailing zeros following the decimal point will be trimmed from the string
Pavyzdžiai	• `format_number(10000000.332,2)` → ‚10,000,000.33‘ if e.g. the current locale is an English variant • `format_number(10000000.332,2,'fr')` → ‚10 000 000,33‘

left

Returns a substring that contains the *n* leftmost characters of the string.

Sintaksė	left(string, length)
Argumentai	• **string** - a string • **length** - integer. The number of characters from the left of the string to return.
Pavyzdžiai	• `left('Hello World',5)` → ‚Hello‘

Further reading: *right*

length

Returns the number of characters in a string or the length of a geometry linestring.

String variant

Returns the number of characters in a string.

Sintaksė	length(string)
Argumentai	• **string** - string to count length of
Pavyzdžiai	• `length('hello')` → 5

Geometry variant

Calculate the length of a geometry line object. Calculations are always planimetric in the Spatial Reference System (SRS) of this geometry, and the units of the returned length will match the units for the SRS. This differs from the calculations performed by the $length function, which will perform ellipsoidal calculations based on the project's ellipsoid and distance unit settings.

Sintaksė	length(geometry)
Argumentai	• **geometry** - line geometry object
Pavyzdžiai	• `length(geom_from_wkt('LINESTRING(0 0, 4 0)'))` → 4.0

lower

Converts a string to lower case letters.

Sintaksė	lower(string)
Argumentai	• **string** - the string to convert to lower case
Pavyzdžiai	• `lower('HELLO World')` → ‚hello world‘

Further reading: *upper*, *title*

lpad

Returns a string padded on the left to the specified width, using a fill character. If the target width is smaller than the string's length, the string is truncated.

Sintaksė	lpad(string, width, fill)
Argumentai	• **string** - string to pad • **width** - length of new string • **fill** - character to pad the remaining space with
Pavyzdžiai	• `lpad('Hello', 10, 'x')` → ‚xxxxxHello‘ • `lpad('Hello', 3, 'x')` → ‚Hel‘

Further reading: *rpad*

ltrim

Removes the longest string containing only the specified characters (a space by default) from the start of string.

Sintaksė	ltrim(string, [characters=' ']) [] žymi neprivalomus argumentus
Argumentai	• **string** - string to trim • **characters** - characters to trim
Pavyzdžiai	• `ltrim(' hello world ')` → ‚hello world ‚ • `ltrim('zzzytest', 'xyz')` → ‚test‘

Further reading: *rtrim*, *trim*

regexp_match

Return the first matching position matching a regular expression within an unicode string, or 0 if the substring is not found.

Sintaksė	regexp_match(input_string, regex)
Argumentai	• **input_string** - the string to test against the regular expression • **regex** - The regular expression to test against. Backslash characters must be double escaped (e.g., „\\s“ to match a white space character or „\\b“ to match a word boundary).
Pavyzdžiai	• `regexp_match('QGIS ROCKS','\\sROCKS')` → 5 • `regexp_match('Budač','udač\\b')` → 2

Further reading: *strpos*

regexp_replace

Returns a string with the supplied regular expression replaced.

Sintaksė	regexp_replace(input_string, regex, replacement)
Argumentai	• **input_string** - the string to replace matches in • **regex** - The regular expression to replace. Backslash characters must be double escaped (e.g., „\\s“ to match a white space character). • **replacement** - The string that will replace any matching occurrences of the supplied regular expression. Captured groups can be inserted into the replacement string using \\1, \\2, etc.
Pavyzdžiai	• `regexp_replace('QGIS SHOULD ROCK','\\sSHOULD\\s',' DOES ')` → ‚QGIS DOES ROCK‘ • `regexp_replace('ABC123','\\d+','')` → ‚ABC‘ • `regexp_replace('my name is John','(.*) is (.*)','\\2 is \\1')` → ‚John is my name‘

Further reading: *replace*, *array_replace*

regexp_substr

Returns the portion of a string which matches a supplied regular expression.

Sintaksė	regexp_substr(input_string, regex)
Argumentai	• **input_string** - the string to find matches in • **regex** - The regular expression to match against. Backslash characters must be double escaped (e.g., „\\s“ to match a white space character).
Pavyzdžiai	• `regexp_substr('abc123','(\\d+)')` → ‚123‘

Further reading: *substr*, *regexp_matches*

repeat

Repeats a string a specified number of times.

Sintaksė	repeat(string, number)
Argumentai	• **string** - string to repeat • **number** - number of times to repeat string
Pavyzdžiai	• `repeat('Hello', 3)` → ‚HelloHelloHello‘

replace

Returns a string with the supplied string, array, or map of strings replaced.

String & array variant

Returns a string with the supplied string or array of strings replaced by a string or an array of strings.

Sintaksė	replace(string, before, after)
Argumentai	• **string** - the input string • **before** - the string or array of strings to replace • **after** - the string or array of strings to use as a replacement
Pavyzdžiai	• `replace('QGIS SHOULD ROCK','SHOULD','DOES')` → ‚QGIS DOES ROCK‘ • `replace('QGIS ABC',array('A','B','C'),array('X','Y','Z'))` → ‚QGIS XYZ‘ • `replace('QGIS',array('Q','S'),'')` → ‚GI‘

Map variant

Returns a string with the supplied map keys replaced by paired values. Longer map keys are evaluated first.

Sintaksė	replace(string, map)
Argumentai	• **string** - the input string • **map** - the map containing keys and values
Pavyzdžiai	• `replace('APP SHOULD ROCK',map('APP','QGIS','SHOULD', 'DOES'))` → ‚QGIS DOES ROCK‘ • `replace('forty two',map('for','4','two','2','forty two', '42'))` → ‚42‘

Further reading: *regexp_replace*, *array_replace*

reverse

Reverses the direction of a line string or reverses a string of text.

String variant

Reverses the order of characters in a string.

Sintaksė	reverse(string)
Argumentai	• **string** - string to reverse
Pavyzdžiai	• `reverse('hello')` → ‚olleh‘

Geometry variant

Reverses the direction of a line string by reversing the order of its vertices.

Sintaksė	reverse(geometry)
Argumentai	• **geometry** - a geometry
Pavyzdžiai	• `geom_to_wkt(reverse(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → ‚LINESTRING(2 2, 1 1, 0 0)‘

right

Returns a substring that contains the *n* rightmost characters of the string.

Sintaksė	right(string, length)
Argumentai	• **string** - a string • **length** - integer. The number of characters from the right of the string to return.
Pavyzdžiai	• `right('Hello World',5)` → ‚World‘

Further reading: *left*

rpad

Returns a string padded on the right to the specified width, using a fill character. If the target width is smaller than the string's length, the string is truncated.

Sintaksė	rpad(string, width, fill)
Argumentai	• **string** - string to pad • **width** - length of new string • **fill** - character to pad the remaining space with
Pavyzdžiai	• `rpad('Hello', 10, 'x')` → ‚Helloxxxxx' • `rpad('Hello', 3, 'x')` → ‚Hel'

Further reading: *lpad*

rtrim

Removes the longest string containing only the specified characters (a space by default) from the end of string.

Sintaksė	rtrim(string, [characters=' ,]) [] žymi neprivalomus argumentus
Argumentai	• **string** - string to trim • **characters** - characters to trim
Pavyzdžiai	• `rtrim(' hello world ')` → , hello world' • `rtrim('testxxzx', 'xyz')` → ‚test'

Further reading: *ltrim*, *trim*

strpos

Return the first matching position of a substring within another string, or 0 if the substring is not found.

Sintaksė	strpos(haystack, needle)
Argumentai	• **haystack** - string that is to be searched • **needle** - string to search for
Pavyzdžiai	• `strpos('HELLO WORLD','WORLD')` → 7 • `strpos('HELLO WORLD','GOODBYE')` → 0

Further reading: *regexp_match*

substr

Returns a part of a string.

Sintaksė	substr(string, start, [length]) [] žymi neprivalomus argumentus
Argumentai	• **string** - the full input string • **start** - integer representing start position to extract beginning with 1; if start is negative, the return string will begin at the end of the string minus the start value • **length** - integer representing length of string to extract; if length is negative, the return string will omit the given length of characters from the end of the string
Pavyzdžiai	• `substr('HELLO WORLD',3,5)` → ‚LLO W‘ • `substr('HELLO WORLD',6)` → ‚ WORLD‘ • `substr('HELLO WORLD',-5)` → ‚WORLD‘ • `substr('HELLO',3,-1)` → ‚LL‘ • `substr('HELLO WORLD',-5,2)` → ‚WO‘ • `substr('HELLO WORLD',-5,-1)` → ‚WORL‘

Further reading: *regexp_substr*, *regexp_matches*

title

Converts all words of a string to title case (all words lower case with leading capital letter).

Sintaksė	title(string)
Argumentai	• **string** - the string to convert to title case
Pavyzdžiai	• `title('hello WOrld')` → ‚Hello World‘

Further reading: *lower*, *upper*

to_string

Converts a number to string. The conversion is not locale-aware, see ‚format_number‘ for a locale-aware alternative.

Sintaksė	to_string(number)
Argumentai	• **number** - Integer or real value. The number to convert to string.
Pavyzdžiai	• `to_string(1.23)` → ‚1.23‘

Further reading: *format_number*

trim

Removes all leading and trailing whitespace (spaces, tabs, etc) from a string.

Sintaksė	trim(string)
Argumentai	• **string** - string to trim
Pavyzdžiai	• `trim(' hello world ')` → ‚hello world‘

Further reading: *ltrim*, *rtrim*

upper

Converts a string to upper case letters.

Sintaksė	upper(string)
Argumentai	• **string** - the string to convert to upper case
Pavyzdžiai	• `upper('hello WOrld')` → ‚HELLO WORLD‘

Further reading: *lower*, *title*

wordwrap

Returns a string wrapped to a maximum/minimum number of characters.

Sintaksė	wordwrap(string, wrap_length, [delimiter_string]) [] žymi neprivalomus argumentus
Argumentai	• **string** - the string to be wrapped • **wrap_length** - an integer. If wrap_length is positive the number represents the ideal maximum number of characters to wrap; if negative, the number represents the minimum number of characters to wrap. • **delimiter_string** - Optional delimiter string to wrap to a new line.
Pavyzdžiai	• `wordwrap('UNIVERSITY OF QGIS',13)` → ‚UNIVERSITY OF QGIS‘ • `wordwrap('UNIVERSITY OF QGIS',-3)` → ‚UNIVERSITY OF QGIS‘

9.2.27 User Expressions

This group contains the expressions saved as *user expressions*.

9.2.28 Variables

This group contains dynamic variables related to the application, the project file and other settings. The availability of variables depends on the context:

- from the Select by expression dialog
- from the Field calculator dialog
- from the layer properties dialog
- from the print layout

To use these variables in an expression, they should be preceded by the `@` character (e.g, `@row_number`).

Variable	Aprašymas
algorithm_id	The unique ID of an algorithm
animation_end_time	End of the animation's overall temporal time range (as a datetime value)
animation_interval	Duration of the animation's overall temporal time range (as an interval value)
animation_start_time	Start of the animation's overall temporal time range (as a datetime value)
atlas_feature	The current atlas feature (as feature object)
atlas_featureid	The current atlas feature ID
atlas_featurenumber	The current atlas feature number in the layout
atlas_filename	The current atlas file name
atlas_geometry	The current atlas feature geometry
atlas_layerid	The current atlas coverage layer ID
atlas_layername	The current atlas coverage layer name
atlas_pagename	The current atlas page name
atlas_totalfeatures	The total number of features in atlas
band	The number of the band in the raster layer
band_description	The description of the band in the raster layer
band_name	The name of the band in the raster layer
canvas_cursor_point	The last cursor position on the canvas in the project's geographical coordinates
cluster_color	The color of symbols within a cluster, or NULL if symbols have mixed colors
cluster_size	The number of symbols contained within a cluster
current_feature	The feature currently being edited in the attribute form or table row
current_geometry	The geometry of the feature currently being edited in the form or the table row
current_parent_feature	represents the feature currently being edited in the parent form. Only usable in an embedded form context.
current_parent_geometry	represents the geometry of the feature currently being edited in the parent form. Only usable in an embedded form context.
form_mode	What the form is used for, like AddFeatureMode, SingleEditMode, MultiEditMode, SearchMode, AggregateSearchMode or IdentifyMode as string.
feature	The current feature being evaluated. This can be used with the ‚attribute' function to evaluate attribute values from the current feature.
frame_duration	Temporal duration of each animation frame (as an interval value)
frame_number	Current frame number during animation playback
frame_rate	Number of frames per second during animation playback
fullextent_maxx	Maximum x value from full canvas extent (including all layers)
fullextent_maxy	Maximum y value from full canvas extent (including all layers)
fullextent_minx	Minimum x value from full canvas extent (including all layers)
fullextent_miny	Minimum y value from full canvas extent (including all layers)

continues on next page

Table 9.1 – tęsinys iš praeito puslapio

Variable	Aprašymas
geometry	The geometry of the current feature being evaluated
geometry_part_count	The number of parts in rendered feature's geometry
geometry_part_num	The current geometry part number for feature being rendered
geometry_point_count	The number of points in the rendered geometry's part
geometry_point_num	The current point number in the rendered geometry's part
geometry_ring_num	Current geometry ring number for feature being rendered (for polygon features only). The exterior ring has a value of 0.
grid_axis	The current grid annotation axis (eg, ‚x' for longitude, ‚y' for latitude)
grid_number	The current grid annotation value
id	The ID of the current feature being evaluated
item_id	The layout item user ID (not necessarily unique)
item_uuid	The layout item unique ID
layer	The current layer
layer_crs	The Coordinate Reference System Authority ID of the current layer
layer_crs_ellipsoid	The ellipsoid Authority ID of the current layer CRS
layer_cursor_point	Point geometry under the mouse position in map canvas (or the GetFeatureInfo position in context of QGIS Server), in active layer's CRS
layer_id	The ID of current layer
layer_ids	The IDs of all the map layers in the current project as a list
layer_name	The name of current layer
layer_vertical_crs	The Identifier for the vertical coordinate reference system of the layer (e.g., ‚EPSG:5703')
layer_vertical_crs_definition	The full definition of the vertical Coordinate reference system of the layer
layer_vertical_crs_description	The name of the vertical Coordinate reference system of the layer
layer_vertical_crs_wkt	The WKT definition of the vertical Coordinate reference system of the current layer
layers	All the map layers in the current project as a list
layout_dpi	The composition resolution (DPI)
layout_name	The layout name
layout_numpages	The number of pages in the layout
layout_page	The page number of the current item in the layout
layout_pageheight	The active page height in the layout (in mm for standard paper sizes, or whatever unit was used for custom paper size)
layout_pageoffsets	Array of Y coordinate of the top of each page. Allows to dynamically position items on pages in a context where page sizes may change
layout_pagewidth	The active page width in the layout (in mm for standard paper sizes, or whatever unit was used for custom paper size)
legend_column_count	The number of columns in the legend
legend_filter_by_map	Indicates if the content of the legend is filtered by the map
legend_filter_out_atlas	Indicates if the atlas is filtered out of the legend
legend_split_layers	Indicates if layers can be split in the legend
legend_title	The title of the legend
legend_wrap_string	The character(s) used to wrap the legend text
map_crs	The Coordinate reference system of the current map
map_crs_acronym	The acronym of the Coordinate reference system of the current map
map_crs_definition	The full definition of the Coordinate reference system of the current map
map_crs_description	The name of the Coordinate reference system of the current map
map_crs_ellipsoid	The acronym of the ellipsoid of the Coordinate reference system of the current map
map_crs_proj4	The Proj4 definition of the Coordinate reference system of the current map
map_crs_projection	The descriptive name of the projection method used by the Coordinate reference system of the map (e.g. ‚Albers Equal Area')
map_crs_wkt	The WKT definition of the Coordinate reference system of the current map
map_end_time	The end of the map's temporal time range (as a datetime value)
map_extent	The geometry representing the current extent of the map

continues on next page

Table 9.1 – tęsinys iš praeito puslapio

Variable	Aprašymas
map_extent_center	The point feature at the center of the map
map_extent_height	The current height of the map
map_extent_width	The current width of the map
map_id	The ID of current map destination. This will be ‚canvas' for canvas renders, and the item ID for layout map renders
map_interval	The duration of the map's temporal time range (as an interval value)
map_layer_ids	The list of map layer IDs visible in the map
map_layers	The list of map layers visible in the map
map_rotation	The current rotation of the map
map_scale	The current scale of the map
map_start_time	The start of the map's temporal time range (as a datetime value)
map_units	The units of map measurements
map_z_range_lower	Lower elevation of the map's elevation range
map_z_range_upper	Upper elevation of the map's elevation range
model_path	Full path (including file name) of current model (or project path if model is embedded in a project).
model_folder	Folder containing current model (or project folder if model is embedded in a project).
model_name	Name of current model
model_group	Group for current model
notification_message	Content of the notification message sent by the provider (available only for actions triggered by provider notifications).
parent	Refers to the current feature in the parent layer, providing access to its attributes and geometry when filtering an *aggregate* function
plot_axis	The associated plot axis, e.g. ‚x' or ‚y'.
plot_axis_value	The current value for the plot axis.
project_abstract	The project abstract, taken from project metadata
project_area_units	The area unit for the current project, used when calculating areas of geometries
project_author	The project author, taken from project metadata
project_basename	The basename of current project's filename (without path and extension)
project_creation_date	The project creation date, taken from project metadata
project_crs	Identifier for the coordinate reference system of the project (e.g., ‚EPSG:4326')
project_crs_arconym	The acronym of the Coordinate reference system of the project
project_crs_definition	The full definition of the Coordinate reference system of the project
project_crs_description	The description of the Coordinate reference system of the project
project_crs_ellipsoid	The ellipsoid of the Coordinate reference system of the project
project_crs_proj4	The Proj4 representation of the Coordinate reference system of the project
project_crs_wkt	The WKT (well known text) representation of the coordinate reference system of the project
project_distance_units	The distance unit for the current project, used when calculating lengths of geometries and distances
project_ellipsoid	The name of the ellipsoid of the current project, used when calculating geodetic areas or lengths of geometries
project_filename	The filename of the current project
project_folder	The folder of the current project
project_home	The home path of the current project
project_identifier	The project identifier, taken from the project's metadata
project_keywords	The project keywords, taken from the project's metadata
project_last_saved	Date/time when project was last saved.
project_path	The full path (including file name) of the current project
project_title	The title of current project
project_units	The units of the project's CRS
project_vertical_crs	Identifier for the vertical Coordinate reference system of the project (e.g., ‚EPSG:5703')
project_vertical_crs_definition	The full definition of the vertical coordinate reference system of the project

continues on next page

Table 9.1 – tęsinys iš praeito puslapio

Variable	Aprašymas
project_vertical_crs_description	The description of the vertical coordinate reference system of the project
project_vertical_crs_wkt	The WKT (well known text) representation of the vertical coordinate reference system of the project
qgis_locale	The current language of QGIS
qgis_os_name	The current Operating system name, eg ‚windows', ‚linux' or ‚osx'
qgis_platform	The QGIS platform, eg ‚desktop' or ‚server'
qgis_release_name	The current QGIS release name
qgis_short_version	The current QGIS version short string
qgis_version	The current QGIS version string
qgis_version_no	The current QGIS version number
row_number	Stores the number of the current row
snapping_results	Gives access to snapping results while digitizing a feature (only available in add feature)
scale_value	The current scale bar distance value
selected_file_path	Selected file path from file widget selector when uploading a file with an external storage system
symbol_angle	The angle of the symbol used to render the feature (valid for marker symbols only)
symbol_color	The color of the symbol used to render the feature
symbol_count	The number of features represented by the symbol (in the layout legend)
symbol_frame	The frame number (for animated symbols only)
symbol_id	The Internal ID of the symbol (in the layout legend)
symbol_label	The label for the symbol (either a user defined label or the default autogenerated label - in the layout legend)
symbol_layer_count	Total number of symbol layers in the symbol
symbol_layer_index	Current symbol layer index
symbol_marker_column	Column number for marker (valid for point pattern fills only).
symbol_marker_row	Row number for marker (valid for point pattern fills only).
user_account_name	The current user's operating system account name
user_full_name	The current user's operating system user name
value	The current value
vector_tile_zoom	Exact vector tile zoom level of the map that is being rendered (derived from the current map scale). Normally in interval [0, 20]. Unlike @zoom_level, this variable is a floating point value which can be used to interpolate values between two integer zoom levels.
with_variable	Allows setting a variable for usage within an expression and avoid recalculating the same value repeatedly
zoom_level	Vector tile zoom level of the map that is being rendered (derived from the current map scale). Normally in interval [0, 20].

Some examples:

- Return the X coordinate of a map item center in layout:

```
x( map_get( item_variables( 'map1'), 'map_extent_center' ) )
```

- Return, for each feature in the current layer, the number of overlapping airport features:

```
aggregate( layer:='airport', aggregate:='count', expression:="code",
              filter:=intersects( $geometry, geometry( @parent ) ) )
```

- Get the object_id of the first snapped point of a line:

```
with_variable(
  'first_snapped_point',
  array_first( @snapping_results ),
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
  attribute(
    get_feature_by_id(
      map_get( @first_snapped_point, 'layer' ),
      map_get( @first_snapped_point, 'feature_id' )
    ),
    'object_id'
  )
)
```

9.2.29 Recent Functions

This group contains recently used functions. Depending on the context of its usage (feature selection, field calculator, generic), recently applied expressions are added to the corresponding list (up to ten expressions), sorted from more to less recent. This makes it easy to quickly retrieve and reapply previously used expressions.

SKYRIUS 10

Stilių biblioteka

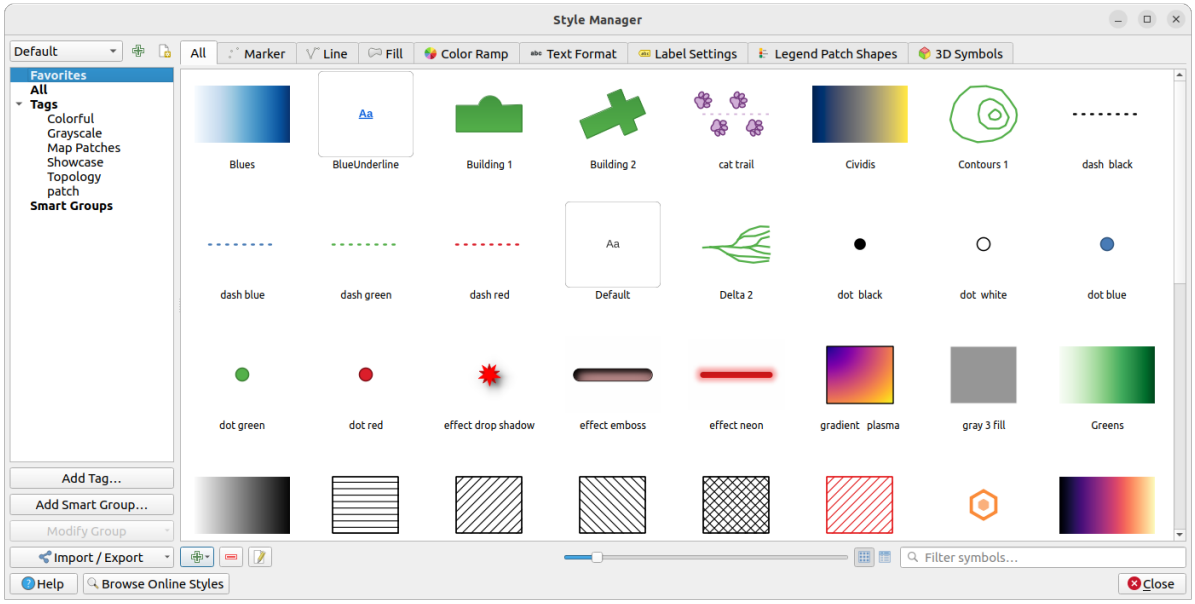
10.1 Stilių tvarkyklė

10.1.1 Stilių tvarkyklės dialogas

Stilių tvarkyklė - tai vieta, kur jūs galite valdyti ir kurti bendrus stilių elementus. Tai yra simboliai, spalvų rampos, teksto formatai ar užrašų nustatymai, kuriuos galite naudoti simbolizuojant geoobjektus, sluoksnius ar spausdinimo išdėstymus. Jie saugomi duomenų bazėje `symbology-style.db`, kuri yra aktyvaus *naudotojo profilyje* ir ji bendra visiems projektams, atveriamiems su tuo profiliu. Stilių elementus galima bendrinti su kitais, naudojant eksportavimo/importavimo savybes *Stilių tvarkyklės* dialoge.

Jūs galite atverti nemodalinį dialogą:

- per meniu *Nustatymai* ► *Stilių tvarkyklė…*
- projekto įrankinės mygtuku Stilių tvarkyklė
- arba mygtuku Stilių tvarkyklė iš meniu *Sluoksnio savybės* ► (kai *konfigūruojate simbolį* ar *formatuojate tekstą*).



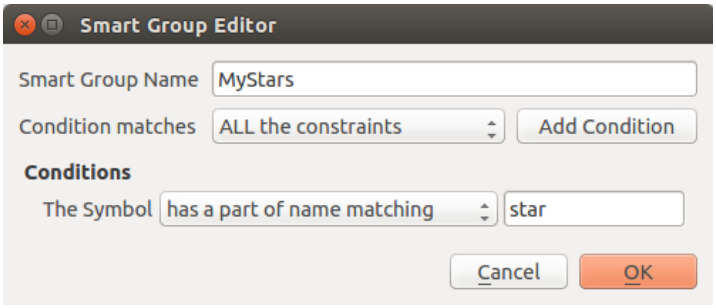
10.1 Fig. : Stilių tvarkyklė

Stiliaus elementų organizavimas

Viršutiniame kairiame *Stilių tvarkyklės* dialogo kampe, iškrentančiame meniu, jūs galite pasirinkti stilių duomenų bazę, prie kurios norite prisijungti. Jei pasirenkate *Numatytoji*, jūs būsite prijungti prie numatytosios stilių duomenų bazės, kurioje rasite visus pagal nutylėjimą prieinamus stilių elementus ir tuos, kuriuos jūs įrašėte į šią duomenų bazę. Pasirinkę *Projekto stilius* jūs prisijungsite prie Projekto stilių duomenų bazės, kur rasite tik tuos simbolius, kuriuos jūs įrašėte į šią duomenų bazę. Jei sukūrėte daugiau stilių duomenų bazių, jos bus matomos šiame iškrentančiame meniu. Ten pat yra parinktys, leidžiančios Pridėti esamą stilių duomenų bazę į projektą ir Kurti naują stilių duomenų bazę (daugiau informacijos rasite skyriuje *Stilių duomenų bazė*).

Kiekvienai duomenų bazei jūs galite organizuoti elementus į skirtingas kategorijas, išvardintas skydelyje kairėje:

- **Parankiniai**: pagal nutylėjimą rodomi konfigūruojant elementą, jame rodoma išplečiama elementų aibė;
- **Visi**: rodomi visi galimi aktyvaus tipo elementai;
- **Žymos**: rodo sąrašą užrašų, kuriuos jūs galite naudoti elementų identifikavimui. Elementui galima pridėti daugiau nei vieną žymą. Parinkite sąraše žymą ir kortelės atnaujinamos - jose rodomi tik elementai su žymomis. Norėdami sukurti naują žymą, kurią vėliau galėtumėte pridėti prie elementų, naudokite mygtuką *Pridėti žymą…* arba kontekstiniame meniu parinkite *Pridėti žymą…*;
- **Išmani grupė**: išmani grupė dinamiškai ištraukia simbolius pagal nurodytas sąlygas (žr., Fig. 10.2). Spauskite mygtuką *Pridėti išmanią grupę…*, kad sukurtumėte išmanią grupę. Dialogas leidžia jums įvesti išraišką parenkamų elementų filtravimui (turi konkrečią žymą, turi tekstą pavadinimą ir pan.). Bet koks simbolis, spalvų rampa, teksto formatas ar užrašų nustatymai, kurie tenkina įvestą sąlygą(as) automatiškai pridedami į išmanią grupę.



10.2 Fig. : Išmanios grupės kūrimas

Žymos ir išmanios grupės viena kitos nepašalina: tai tiesiog du skirtingi jūsų stiliaus elementų organizavimo būdai. Priešingai nei išmanios grupės, kurios automatiškai ištraukia jiems priklausančius elementus pagal įvesties apribojimus, žymas pildo naudotojas. Norėdami keisti bet kurią iš šių kategorijų jūs galite:

- parinkti elementus, spausti dešinį pelės mygtuką ir parinkti *Pridėti į žymą* ► ir tada parinkti žymos pavadinimą arba sukurti naują žymą;
- parinkti žymą ir spausti *Keisti grupę…* ► *Pridėti parinktą žymą prie simbolių*. Prie kiekvieno elemento atsiranda varnelė, kurią jūs galite parinkti arba ne. Baigus parinkimą spauskite *Keisti grupę…* ► *Baigti žymėjimą*.
- parinkti išmanią grupę, spausti *Keisti grupę…* ► *Keisti išmanią grupę…* ir konfigūruoti naują sąlygų rinkinį dialoge *Išmanių grupių redaktorius*. Šią parinktį taipogi rasite ir išmanios grupės kontekstiniame meniu.

Norėdami pašalinti žymą ar išmanią grupę, spauskite ją dešiniu pelės mygtuku ir parinkite mygtuką *Išimti*. Pastebėtina, kad tai neištrina elementų, sugrupuotų į kategoriją.

Dialogo *Stiliaus tvarkyklė* centrinėje dalyje yra rėmas su peržiūrėtais elementais, organizuotais kortelėse:

- *Visi* pilnam sąrašui taškų, linijų ir paviršių simbolių bei užrašų nustatymų, o taip pat iš anksto nustatytoms rampoms ir teksto formatams;
- *Žymekliai* tik taškų simboliams;
- *Linijos* tik linijų simboliams;
- *Pildymas* tik paviršiaus simboliams;
- *Spalvų rampa*;
- abc *Teksto formatai* valdo *teksto formatus*, kuriuose laikomas teksto šriftas, spalva, buferiai, šešėliai ir fonas (t.y. visos užrašų nustatymo formatavimo dalys, kurios gali būti naudojamos, pavyzdžiui, išdėstymuose);
- *Užrašų nustatymai* valdo *užrašų nustatymus*, kuriuose yra teksto formatai, kai kuriems sluoksnių tipams būdingi nustatymai, tokie kaip užrašų vietos, prioritetai, išnešimai, braižymas…
- *Legendos elementų formos* valdo savo legendos elementų formas, kuriose yra *Žymeklių*, *Linijų* ir *Pildymo* geometrijos.
- *3D simboliai* konfigūruoja simbolius su *3D savybėmis* (išsikišimai, šešėliai, aukštis…) geoobjektams, kurie braižomi *3D žemėlapių vaizde*

Apatinėje dešinėje pusėje galite parinkti, ar norite išdėlioti stilius *Piktogramų vaizdu* arba *Sąrašo vaizdu*. Abiejuose vaizduose įrankinėje rodomas didesnis stiliaus egzempliorius. Kairėje piktogramų pusėje esantis miniatiūros dydžio slankiklis padeda jums reguliuoti realius miniatiūrų dydžius dialoge, kad galėtumėte geriau peržiūrėti simbolius!

Elemento pridėjimas, keitimas ir pašalinimas

Kaip matoma anksčiau, stilių elementai rodomi įvairiose kortelėse, kurių turinys priklauso nuo aktyvios kategorijos (žymos, išmani grupė, parankiniai…). Kai kortelė įjungta, jūs galite:

- Pridėti naujus elementus: spauskite mygtuką Pridėti elementą ir konfigūruokite elementą sekdami *simbolių*, *spalvų rampų* ar *teksto formatų ir užrašų* kūrimo aprašymus.
- Keisti esamą elementą: parinkite elementą ir spauskite mygtuką Keisti elementą ir konfigūruokite kaip minima aukščiau.
- Trinti esamus elementus: kad ištrintumėte elementą, kurio jums daugiau nebereikia, parinkite jį ir spauskite Išimti elementą (taipogi pasiekiama paspaudus dešinį pelės mygtuką). Elementas bus ištrintas iš vietinės duomenų bazės.

Pastebėtina, kad kortelė *Visi* teikia prieigą prie šitų parinkčių kiekvienam elemento tipui.

Paspaudus parinktus elementus dešinius pelės mygtuku leidžia jums:

- *Pridėti į parankinius*;
- *Išimti iš parankinių*;
- *Pridėti į žymą* ► ir parinkti atitinkamą žymą arba sukurti naują naudojimui; šiuo metu paskirtos žymos yra pažymėtos;
- *Išvalyti žymas*: atjungiant simbolius nuo visų žymų;
- *Išimti elementą(us)*;
- *Keisti elementą*: taikoma elementui, kurį paspaudėte dešiniu pelės mygtuku;
- *Kopijuoti elementą*;
- *Įkelti elementą …*: įkelia į vieną iš stiliaus tvarkyklės kategorijų arba kitur QGIS (simbolių ar spalvų mygtukai)
- *Eksportuoti parinktą simbolį(ius) kaip PNG…* (taikoma tik simboliams);
- *Eksportuoti parinktą simbolį(ius) kaip SVG…* (taikoma tik simboliams);

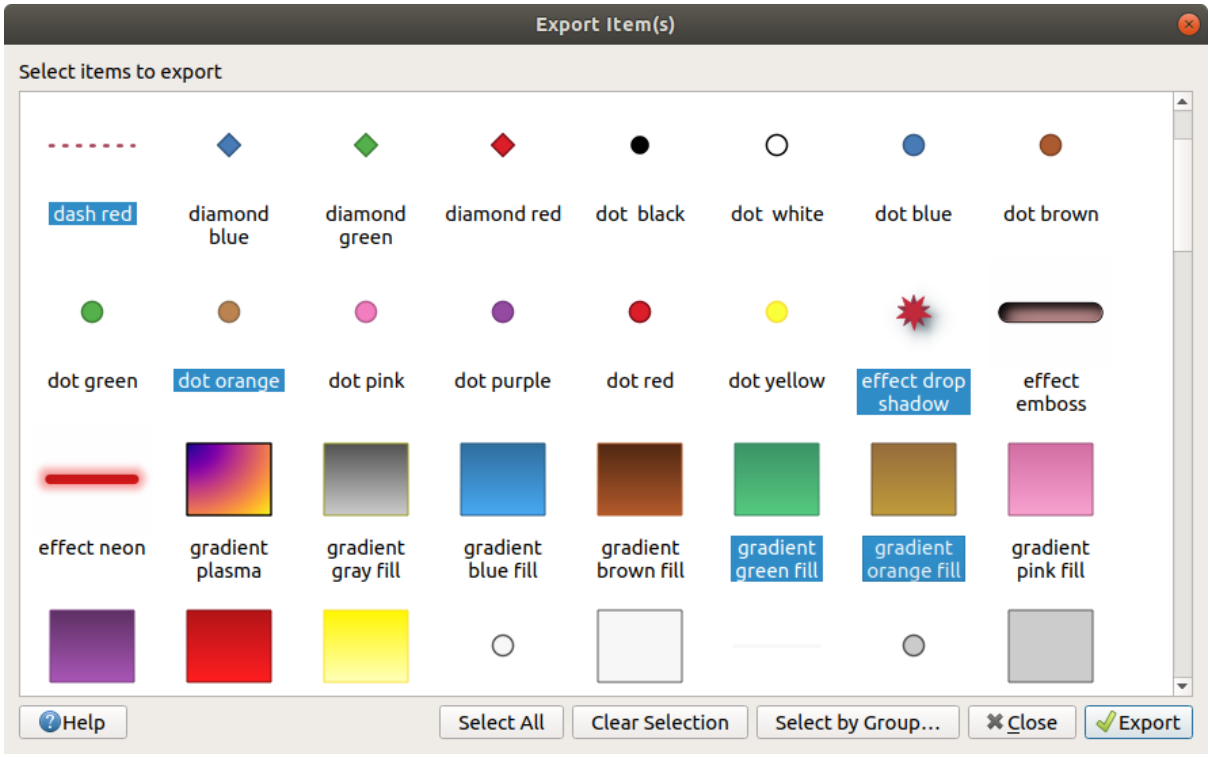
Stiliaus elementų bendrinimas

Stilių tvarkyklės dialogo apatinėje kairėje pusėje esantis įrankis *Importuoti/Eksportuoti* leidžia paprastai bendrinti su kitais simbolius, spalvų rampas, teksto formatus ir užrašų nustatymus. Šios parinktys taipogi pasiekiamos paspaudus dešinį pelės mygtuką ant elementų.

Elementų eksportavimas

Jūs galite eksportuoti aibę elementų į `.XML` failą:

1. Išskleiskite iškrentantį meniu *Importuoti/Eksportuoti* ir parinkite *Eksportuoti elementą(us)…*
2. Parinkite elementus, kuriuos norite integruoti. Parinkimą galima atlikti su pele arba naudojant žymas ar anksčiau sukurtą grupę.
3. Kai būsite pasiruošę - spauskite *Eksportuoti*. Jūsų paprašys nurodyti įrašomo failo vietą. XML formatas sukuria vieną failą, kuriame yra visi parinkti elementai. Šį failą tada galima importuoti į kito naudotojo stilių biblioteką.



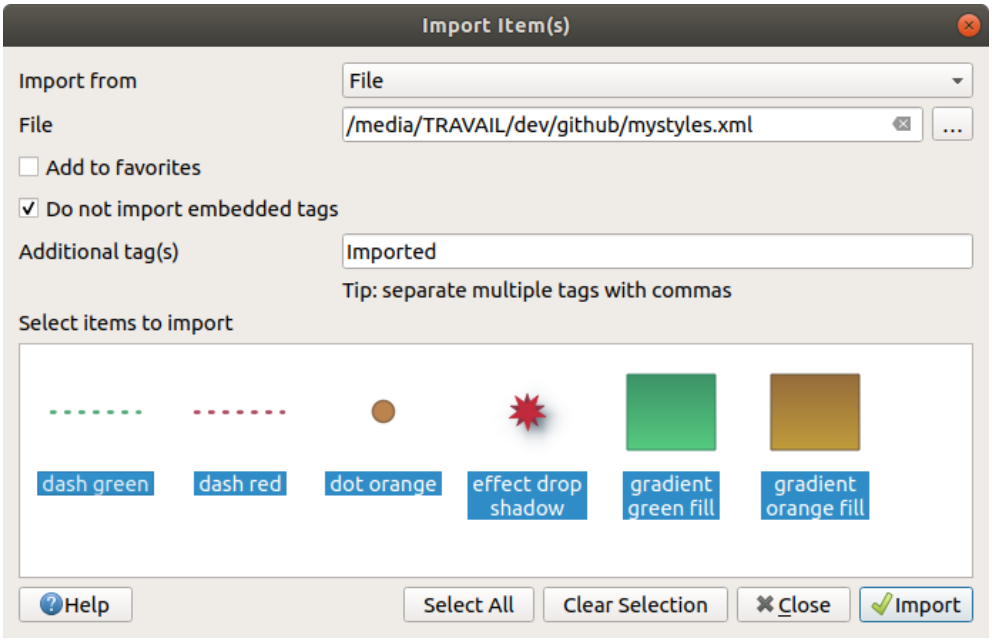
10.3 Fig. : Stiliaus elementų eksportavimas

Parinkus simbolius jūs taipogi galite eksportuoti juos į `.PNG` ar `.SVG`. Eksportuojant į `.PNG` ar `.SVG` (abu negalimi kitiems stilių elementų tipams) nurodytame aplanke sukuriamas failas kiekvienam parinktam simboliui. SVG aplanką galima pridėti į kito naudotojo *SVG kelius* per meniu *Nustatymai* ► *Parinktys* ► *Sistema*, taip jis galės tiesiogiai pasiekti šiuos simbolius.

Elementų importavimas

Jūs galite praplėsti savo stilių biblioteką importuodami naujus elementus:

1. Apatiniame kairiame dialogo kampe išskleiskite iškrentantį meniu *Importuoti/Eksportuoti* ir parinkite *Importuoti elementą(us)*.
2. Naujame dialoge nurodykite stiliaus elementų šaltinį (tai gali būti `.xml` failas diske arba url).
3. Nustatykite, ar reikia importuojamus elementus *Pridėti į parankinius*.
4. Įjunkite parinktį *Neimportuoti įtrauktų žymių*, jei nenorite importuoti su importuojamais elementais susijusių žymų.
5. Nurodykite pavadinimus bet kokiai *Papildoma žymai(oms)*, kurias reikia pritaikyti naujiems elementams.
6. Peržiūroje parinkite simbolius, kuriuos norite pridėti į jūsų biblioteką.
7. Ir spauskite *Importuoti*.



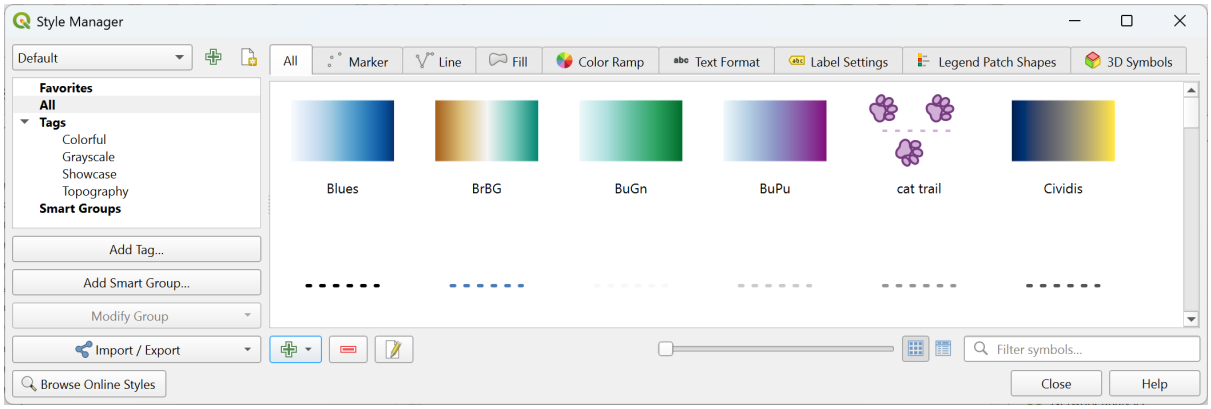
10.4 Fig. : Stiliaus elementų importavimas

Naršymo skydelio naudojimas

Stiliaus elementus taipogi galima importuoti ir tiesiogiai į aktyvaus naudotojo profilio stiliaus duomenų bazę per skydelį *Naršyklė*:

1. Parinkite naršyklėje stiliaus `.xml` failą
2. Nutempkite jį virš žemėlapio drobės arba spauskite dešinį pelės mygtuką ir parinkite *Importuoti stilių…*
3. Užpildykite dialogą *Importuoti elementus* pagal *Elementų importavimas*
4. Spauskite *Importuoti* ir parinkti stiliaus elementai bus pridėti į stiliaus duomenų bazę

Naršyklėje du kartus paspaudus stiliaus failą atsidaro dialogas *Stilių tvarkyklė*, kuriame rodomi faile esantys elementai. Jūs galite parinkti juos ir spausti *Kopijuoti į numatytąjį stilių…*, kad importuotumėte į aktyvią stiliaus duomenų bazę. Elementams galima priskirti žymas. Tai taip pat pasiekiama ir paspaudus dešinį pelės mygtuką bei parinkus komandą *Atverti stilių…*.



10.5 Fig. : Stiliaus elementų failo atidarymas

Dialogas taipogi leidžia eksportuoti vieną simbolį kaip `.PNG` ar `.SVG` failą.

Internetinės repozitorijos naudojimas

QGIS projektas palaiko repozitoriją su rinkiniu stilių, kuriuos bendrina QGIS naudotojai. Jos adresas yra https://hub.qgis.org/styles ir ją galima pasiekti per dialogą *Stilių tvarkyklė*, paspaudus apačioje esantį mygtuką *Naršyti stilius internete*.

Toje repozitorijoje jūs galite:

1. Naršyti ir ieškoti bet kokių stiliaus elementų pagal jų tipą ar pavadinimą
2. Atsisiųsti stiliaus failą ir jį išpakuoti
3. Įkelti `.xml` paremtą failą į jūsų QGIS stiliaus duomenų bazę naudojant bet kurį iš aukščiau minėtų būdų.

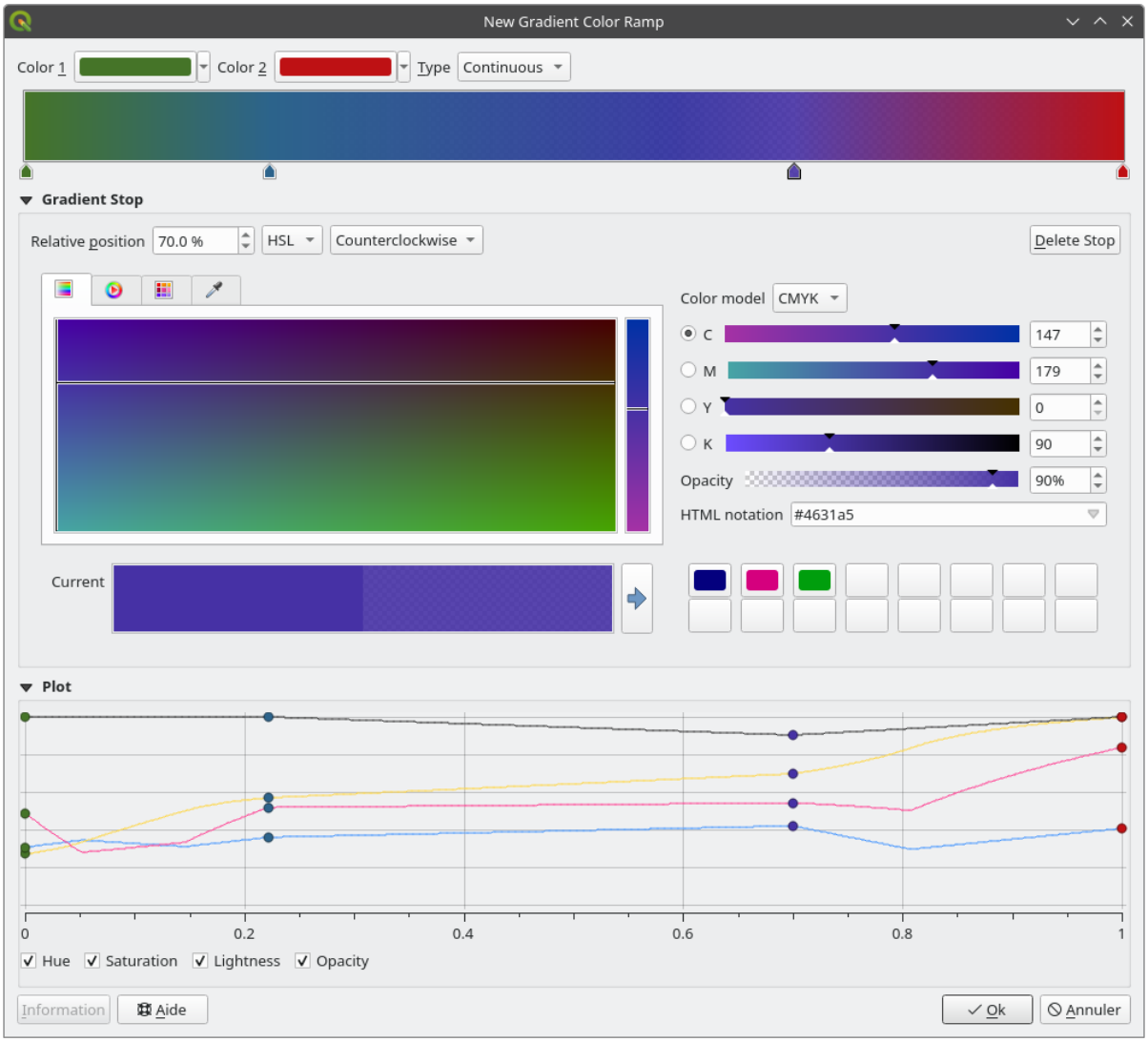
10.1.2 Spalvų rampos kūrimas

Stilių tvarkyklės dialogo spalvų rampos kortelė padeda jums peržiūrėti skirtingas spalvų rampas pagal skydelio kairėje parinktą kategoriją.

Norėdami sukurti savo spalvų rampą, aktyvuokite Spalvų rampos kortelę ir spauskite mygtuką Pridėti elementą. Mygtukas atskleidžia iškrentantį sąrašą, kuriame jūs turite pasirinkti rampos tipą:

- *Gradientas*: pagal pradžios ir pabaigos spalvas sukuriama spalvų rampa, kuri gali būti **tolydi** ar **diskreti**. Du kartus paspaudus rampos peržiūroje jūs galite pridėti tiek tarpinių spalvų, kiek norite. Spauskite spalvos sustojimo indikatorių ir *Gradiento sustojime* jūs galėsite:
  - keisti jo *Santykinę poziciją* nuo spalvų rampos pradžios. Taip pat galima ir tempti indikatorių pele arba spausti rodyklės mygtukus (kartu su `Shift`, jei norite keisti greičiau)
  - nurodyti spalvų modelį, kurį reikia naudoti interpoliuojant tarp spalvų: tai gali būti *RGB*, *HSL* ar *HSV*. Kai kuriais atvejais ši parinktis gali padėti išvengti nesodrių tonų, taip gaunant akiai patrauklesnius gradientus.
  - nustatyti kryptį, kurią turi sekti interpoliacija *HSL* ar *HSV* spalvos specifikacijos **Spalvos** komponentui. Tai gali būti *Pagal laikrodžio rodyklę* arba *Prieš laikrodžio rodyklę*.
  - nustatyti *spalvų savybes*
  - išimti spalvų sustojimą paspaudus *Trinti sustojimą* arba `DEL`

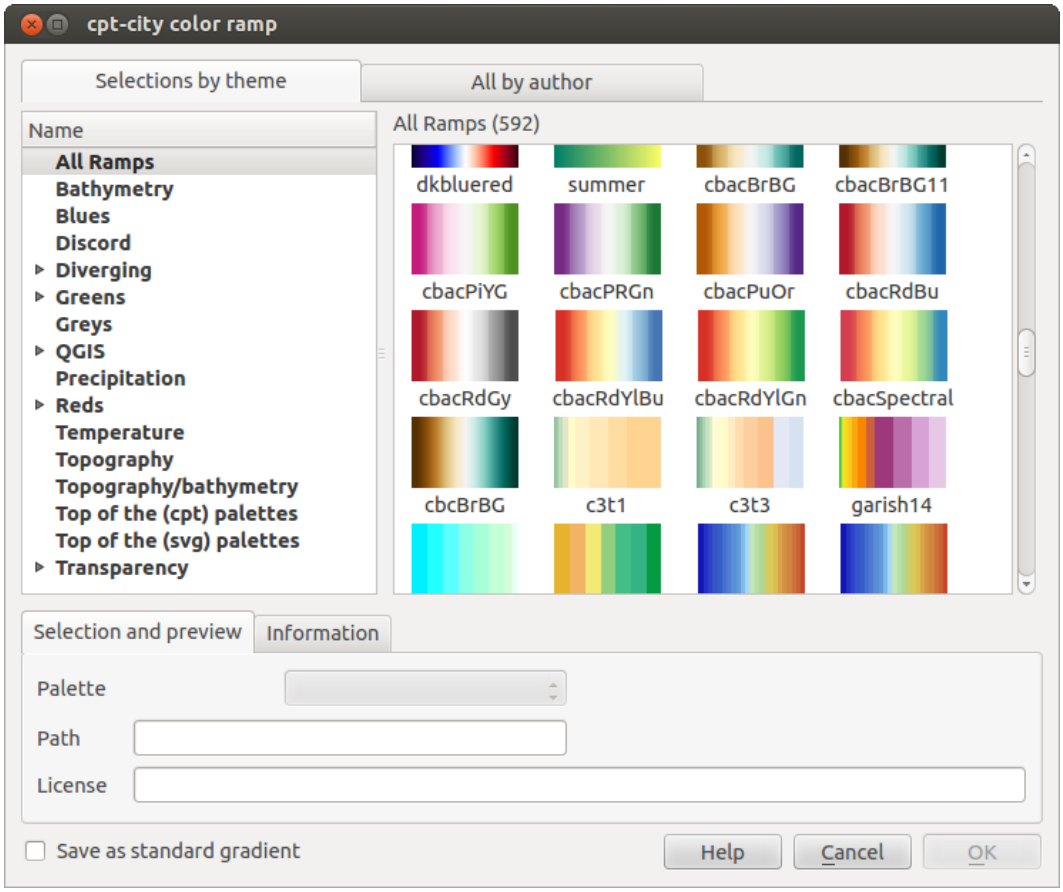
  Grupė *Braižymas* teikia kitą grafinį būdą kurti spalvų rampą, keičiant spalvų sustojimų vietą ar permatomumą bei HSL komponentus.



10.6 Fig. : Savo gradiento spalvos rampos pavyzdys su keliais sustojimais

> **Patarimas**
>
> Pertempus spalvą iš spalvų paletės į gradiento rampos peržiūrą pridės naują spalvos sustojimą.

- *Standartinės spalvos*: leidžia kurti spalvų rampą, kuri susideda iš naudotojo parinktų spalvų;
- *Atsitiktinis*: sukuria atsitiktinių spalvų aibę pagal *Spalvos*, *Sodrumo*, *Reikšmės* ir *Permatomumo* reikšmių intervalą bei spalvų skaičių (*Klasės*);
- *Katalogas: ColorBrewer*: iš anksto apibrėžtų diskrečių spalvų gradientų aibė, kurioje jūs galite keisti rampos spalvų skaičių;
- ar *Katalogas: cpt-city*: prieiga prie viso spalvų gradientų katalogo, kuriuos galite vietoje *įrašyti kaip standartinius gradientus*. cpt-city parinktis atveria naują dialogą su šimtais „pagal nutylėjimą“ įtrauktų temų.



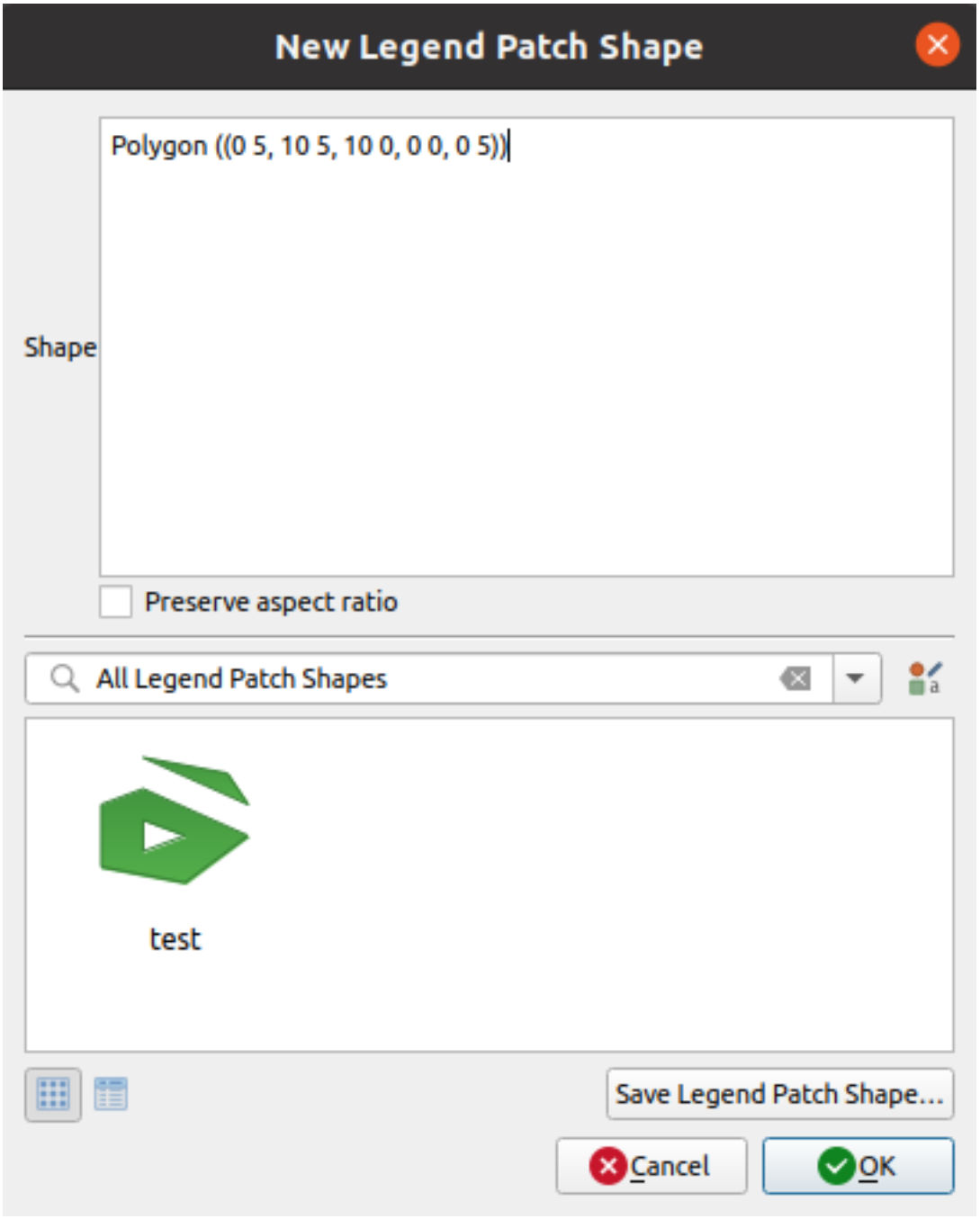
10.7 Fig. : cpt-city dialogas su šimtais spalvų rampų

10.1.3 Legendos elemento formos kūrimas

Norėdami sukurti naują legendos elemento formą, aktyvuokite kortelę *Legendos elemento formos* ir spauskite mygtuką Pridėti elementą. Mygtukas atskleidžia iškrentantį sąrašą, kuriame galite pasirinkti geometrijos tipą:

- *Žymeklių legendos elemento forma…*: naudojimui su taškų geometrijomis.
- *Linijų legendos elementų formos…*: naudojimui su linijų geometrijomis.
- *Pildymo legendos elemento forma…*: naudojimui su poligonų geometrijomis.

Visos trys parinktys rodys tą patį dialogą.

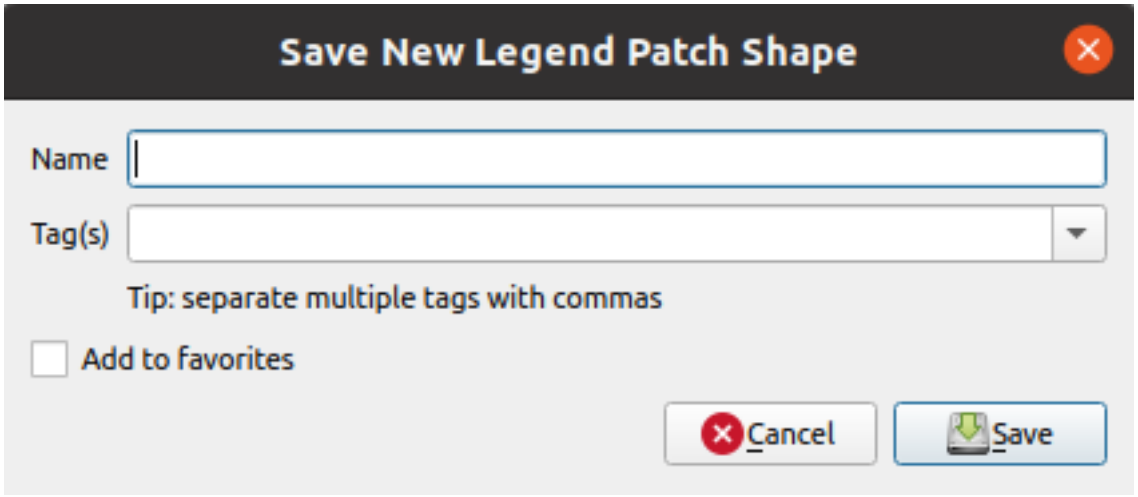


10.8 Fig. : Naujos legendos elemento formos kūrimas

Nuo parinkto geometrijos tipo skirsis tik geometrijos tipas ir rodoma legendos elemento forma. Galimos šios parinktys:

- *Forma*: nurodykite legendos elemento formą kaip WKT tekstą. Galima naudoti tiek vienos, tiek kelių dalių geometrijas, bet negalima naudoti Geometrijų rinkinio.
- *Išlaikyti proporcijas*
- *Piktogramų vaizdas* ar *Sąrašo vaizdas* galimoms legendos elementų formoms, filtruojant pagal žymas.

Kai apibrėžiama nauja forma, jūs galite *Įrašyti legendos elemento formą…* ar spausti *Gerai*, kurie abu ves į tą patį dialogą.



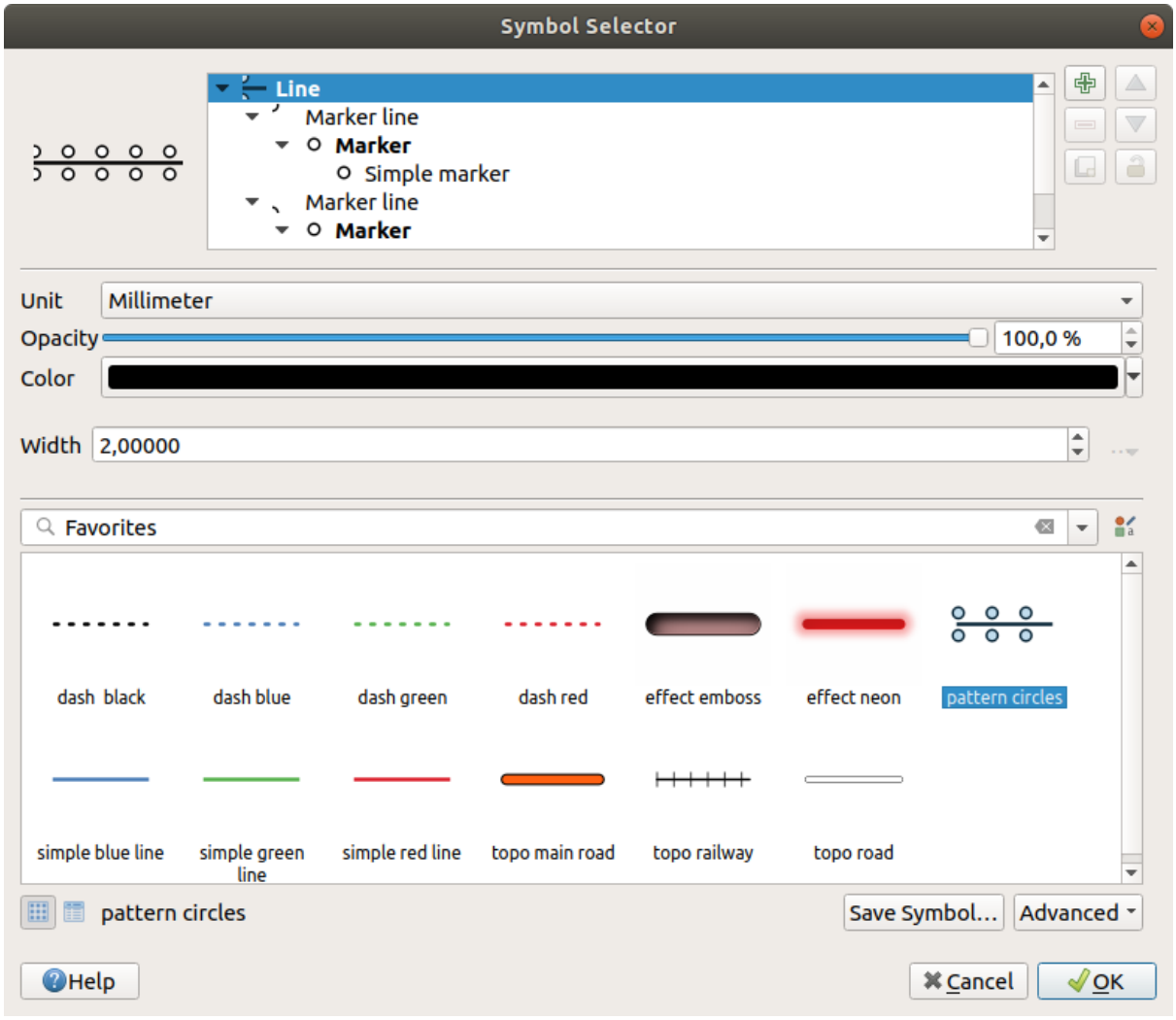
10.9 Fig. : Įrašyti naują legendos elemento formą

Čia jūs galite parinkti pavadinimą, žymas ir aprašyti formą ir ar ji turėtų būti pridėta į parankinius.

Jei paspausite *Įrašyti…*, forma bus pridėta į sąrašą ir jus nuves atgal į dialogą *Nauja legendos elemento forma*, kur galėsite tęsti kurti naujas formas.

10.2 Simbolių parinkiklis

Simbolių parinkiklis yra pagrindinis simbolio kūrimo dialogas. Jūs galite kurti ar keisti žymeklių, linijų ar pildymo simbolius.



10.10 Fig. : Linijos simbolio kūrimas

Simbolio parinkimo dialogą sudaro du pagrindiniai struktūriniai komponentai:

- simbolių medis, rodantis simbolio sluoksnius, kurie vėliau kombinuojami, kad sukurtų naują globalų simbolį
- ir medyje parinkto simbolio sluoksnio konfigūravimo nustatymai.

10.2.1 Simbolio sluoksnių medis

Simbolį gali sudaryti keli *Simbolio sluoksniai*. Simbolio medis rodo šiuos simbolių sluoksnius, kurie vėliau kombinuojami, kad suformuotų naują globalų simbolį. Bet to, simbolio vaizdas dinamiškai atnaujinamas kai tik pakeičiamos simbolio savybės.

Priklausomai nuo simbolio medyje parinkto elemento lygio, galima naudoti įvairius įrankius, leidžiančius valdyti medį:

- pridėti naują simbolio sluoksnį: galite sukrauti kiek tik norite simbolių
- išimti parinktą simbolio sluoksnį
- užrakinti simbolio sluoksnio spalvas: užrakinta spalva nesikeičia, kai naudotojas keičia spalvą globaliame (ar aukštesniame) simbolio lygyje

- kurti simbolio sluoksnio(ių) (grupės) kopiją
- perkelti simbolio sluoksnį aukštyn ar žemyn

10.2.2 Simbolio konfigūravimas

QGIS'e simbolio konfigūravimas atliekamas dviem žingsniais: simbolio ir tada simbolio sluoksnio.

Simbolis

Viršutinis medžio lygis priklauso nuo sluoksnio geometrijos ir gali būti **Žymeklis**, **Linija** arba **Užpildymas**. Kiekvienas simbolis gali įtraukti vieną ar daugiau simbolių (įskaitant ir bet kurio kito tipo) ar simbolio sluoksnių.

Jūs galite konfigūruoti keletą parametrų, kurie taikomi globaliam simboliui:

- *Vienetai*: gali būti **Milimetrai**, **Taškai**, **Pikseliai**, **Metrai pagal mastelį**, **Žemėlapio vienetai** ar **Coliai** (daugiau informacijos rasite *Vienetų parinkiklis*)
- *Nepermatomumas*
- *Spalva*: kai naudotojas pakeičia šį parametrą, jo reikšmė atsispindi visose atrakintų sub-simbolių spalvose
- *Dydis* ir *Pasukimas* žymeklių simboliams
- *Plotis* linijų simboliams

> **Patarimas**
>
> Naudokite *Dydžio* (žymeklių simboliams) ar *Pločio* (linijų simboliams) savybes simbolių sluoksnyje, kad proporcingai keistumėte dydį visuose įtrauktuose *simbolio sluoksnių* matmenyse.

> **Pastaba**
>
> *Nuo duomenų priklausančio permušimo* mygtukas, esantis greta pločio, dydžio ar pasukimo paramtrų yra neaktyvus, kai simbolis nustatomas iš Stilių tvarkyklės dialogo. Kai simbolis prijungiamas prie žemėlapio sluoksnio, šis mygtukas padeda jums sukurti *santykinės ar kelių kintamųjų analizės* braižymą.

- *Simbolių bibliotekos* peržiūra: rodomi to paties tipo simboliai ir per aukščiau keičiamą iškrentantį sąrašą gali būti filtruojami laisvos formos tekstu arba *kategorijomis*. Jūs galite atnaujinti simbolių sąrašą naudodami mygtuką Stilių tvarkyklė ir atverdami eponimų dialogą. Ten jūs galite naudoti bet kokias galimybes, aprašytas skiltyje *Stilių tvarkyklė*.

  Simboliai rodomi arba:

  - piktogramų sąraše (su miniatiūra, pavadinimu ir susijusiomis žymomis) naudojant po sekcija esantį mygtuką Sąrašo vaizdas;
  - arba piktogramų peržiūroje naudojant mygtuką Piktogramų vaizdas.
- Spauskite mygtuką *Įrašyti simbolį*, kad atvertumėte dialogą *Įrašyti naują simbolį*. Čia galite parinkti *Paskirtį*, kur norite pridėti keičiamą simbolį, nurodykite jam *Pavadinimą* ir add *Žymą(s)*. Jūs taipogi galite jūsų naują simbolį Pridėti į parankinius.
- Su parinktimi *Sudėtingesni* , jūs galite:
  - linijų ir pildymo simboliams *Apkarpyti geoobjektus pagal drobės apimtį*.

  - pildymo simboliams *Priverstinai taikyti dešinės rankos orientacijos taisyklę*: leidžia priversti, kad braižomi simboliai atitiktų standartinę „dešinės rankos taisyklę“ žiedų orientacijai (t.y. poligonų išorinis žiedas eina pagal laikrodžio rodyklę, o vidiniai žiedai - prieš laikrodžio rodyklę).

    Orientacija taikoma tik braižymui, pradinė geoobjekto geometrija nekeičiama. Tai leidžia kurti pildymo simbolius su stabilia išvaizda, nepriklausomai nuo braižomo duomenų rinkinio ir konkrečių geoobjektų žiedų orientacijos.
  - žymeklių simboliams *Buferio nustatymai…* aplink žymeklio simbolį įjungia papildomą spindėjimo efektą, taip jį padarant skaitomu virš įvairių fonų. Buferis skaičiuojamas ir braižomas naudojant VISŲ žymeklio simbolio sluoksnių formą. Jūs galite keisti rodymo *Dydį*, *Jungimo stilių* ir *Simbolį*.
  - for all symbol types, enable and configure a distance for the map canvas *Extent buffer*, or the *Animation settings*.
  - Depending on the *symbology* of the layer the symbol is being applied to, additional settings are available in the *Advanced* menu:
    * *Simbolių lygiai…* norint apibrėžti simbolių braižymo rikiuotę
    * *Nuo duomenų priklausanti dydžio legenda*
    * *Atitikti įrašytiems simboliams…* ir *Atitikti simboliams iš failo…* automatiniam *simbolių priskyrimui klasėms*

Simbolių sluoksnis

Žemesniame medžio lygmenyje jūs galite konfigūruoti simbolių sluoksnius. Galimi simbolių sluoksnio tipai priklauso nuo aukštesnio simbolio tipo. Galite taikyti simbolio sluoksnio *paišymo efektus*, kad pagerintumėte jo braižymą. Kadangi aprašyti visų simbolių sluoksnių tipų visas parinktis yra neįmanoma, žemiau aprašome tik kelias svarbias.

Bendri parametrai

Simbolių sluoksnio kūrimui, nepriklausomai nuo jo žymeklių, linijų ar pildymo potipio, teikiamos kelios bendros parinktys ir valdikliai:

- *spalvų parinkimo* valdiklis lengvam spalvų valdymui
- *Vienetai*: gali būti **Milimetrai**, **Taškai**, **Pikseliai**, **Metrai pagal mastelį**, **Žemėlapio vienetai** ar **Coliai** (daugiau informacijos rasite *Vienetų parinkiklis*)
- Nuo duomenų priklausančių permušimų valdiklis prie beveik visų parinkčių, praplečiantis kiekvieno simbolio tinkinimo galimybes (daugiau informacijos rasite skyriuje *Nuo duomenų priklausančių permušimų nustatymai*)
- Parinktis *Įjungti simbolinį sluoksnį* valdo simbolio matomumą. Išjungti simbolių sluoksniai nepaišomi braižant simbolį, bet įrašomi simbolyje. Galimybė paslėpti simbolių sluoksnius patogi, kai ieškote geriausio jūsų simbolio dizaino, bet nenorite nieko ištrinti bandymu metu. Nuo duomenų priklausantis permušimas tada leidžia paslėpti ar rodyti skirtingus simbolių sluoksnius priklausomai nuo išraiškų (pavyzdžiui naudojant geoobjektų atributus).
- *Paišymo efektų* mygtukas *įtakoja braižymą*.

Pastaba

Nors žemiau pateiktas aprašymas daro prielaidą, kad simbolio tipas susijęs su geoobjekto geometrija, turėkite omenyje, kad jūs galite įdėti vienus simbolių sluoksnius į kitus. Tokiu atveju žemesnio lygio simbolio sluoksnio parametras (vieta, poslinkis…) gali būti susijęs su aukštesnio lygio simboliu, o ne su pačia geoobjekto geometrija.

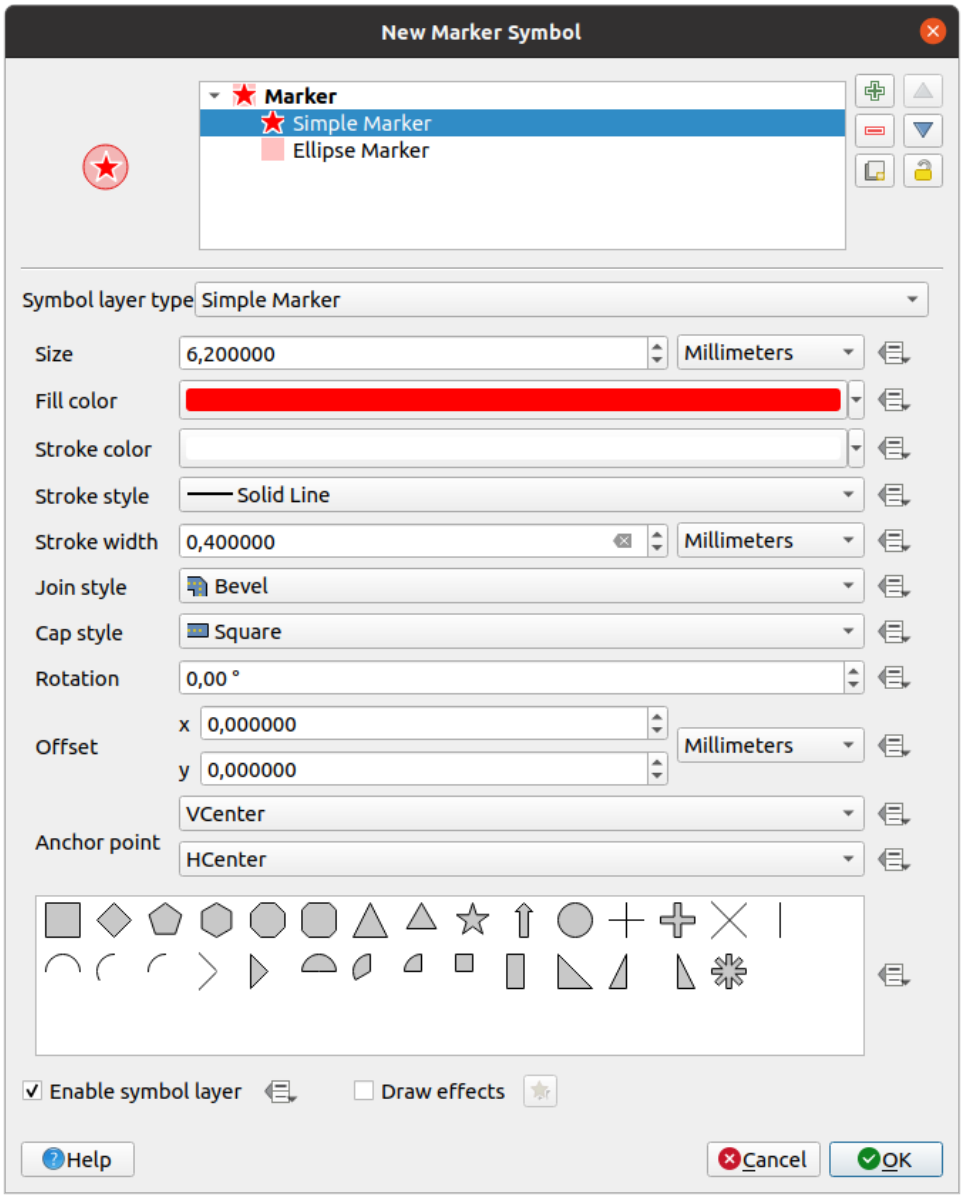
Žymeklių simboliai

Taškų geometrijų geoobjektams tinkami žymeklių simboliai turi kelis *Simbolių sluoksnių tipus*:

10.1 Table : List of marker symbol layer types

Simple marker	*Animated marker*	*Ellipse marker*
Filled marker	*Font marker*	*Geometry generator*
Mask	*Raster image marker*	*Vector Field marker*
SVG marker		

- **Paprastas žymeklis** (numatytasis)



10.11 Fig. : Paprasto žymeklio simbolio kūrimas

Paprasto žymeklio simbolio sluoksnio tipas turi šias savybes:

- *Dydį* įvairiais palaikomais vienetais

  - *Užpildymo spalvą*
  - *Brėžimo spalvą*, *Brėžimo stilių* iš anksto nustatytame sąraše ir *Brėžimo dydį*
  - *Jungimo stilius*: gali būti **Kampas**, **Mitra** ar **Apvalus**
  - *Pabaigos stilius*: gali būti **Kvadratinis**, **Nukirstas** ar **Apvalus**
  - *Pasukimas*
  - *Poslinkis X* ir *Y* kryptimis nuo geoobjekto
  - *Kitas taškas*: nustatantis kvadranto tašką ant simbolio, taip pririšant pozicijos atskaitos tašką. Tai yra taškas, kuriam taikomas *Poslinkis*.

- **Animuotas žymeklis** (žr. *Animuotas žymeklis*)
- **Elipsės žymeklis**: paprastas keičiamo pločio ir aukščio žymeklių simbolių sluoksnis
- **Užpildyti žymekliai**: panašiai kaip paprastas žymeklių sluoksnis, tik jis žymeklių braižymui naudoja *užpildytą sub simbolį*. Tai leidžia naudoti visus esamus QGIS pildymų (ir brėžimų) stilius žymeklių braižymui, pavyzdžiui gradientą ar išdeginimo pildymus.
- **Šrifto žymeklis**: panašiai kaip paprastas žymeklis, tik jis naudoja įdiegtus šriftus žymeklio braižymui. Jo papildomos savybės yra:
  - *Šrifto šeima*
  - *Šrifto stilius*
  - *Simbolis(iai)*, reprezentuojantys tekstą, kurį reikia rodyti kaip simbolį. Juos galima įrašyti arba parinkti iš šriftų simbolių rinkinio valdiklio. Jūs galite *Peržiūrėti* juos su parinktais nustatymais.
- **Geometrijos generatorius** (žr. *Geometrijos generatorius*)
- **Slėpimas**: tai sub-simbolis, kuris apibrėžia slėpimo formą, kurios spalva bus ignoruojama ir naudojamas tik nepermatomumas. Tai patogu, kai žymeklis persidengia su kitų simbolių užrašais su panašiomis spalvomis, dėl ko juos sunku įžiūrėti. Daugiau informacijos galima rasti skyriuje *Masks Properties*.
- **Rastro piešinio žymeklis**: naudoja piešinį (`PNG`, `JPG`, `BMP` ...) kaip žymeklio simbolį. Piešinys gali būti failas diske, nutolęs URL, įtrauktas į stilių duomenų bazę (*daugiau informacijos*) arba gali būti užkoduotas kaip base64 eilutė. Piešinio plotį ir aukštį galima nurodyti nepriklausomai, arba naudojant Išlaikyti proporcijas. Dydį galima nurodyti naudojant bet kurios *bendrus vienetus* arba kaip procentus nuo originalaus piešinio dydžio (pakeitus pagal plotį).
- **Vektorinio lauko žymeklis** (žr. *Vektorinio lauko žymeklis*)
- **SVG žymeklis**: suteikia jums piešinius iš jūsų SVG kelių (nurodytų per meniu *Nustatymai* ► *Parinktys...* ► *Sistema*) žymeklių simbolių braižymui. Simbolių plotį ir aukštį galima nurodyti atskirai arba naudojant Užrakintą proporciją. Kiekvieno SVG failo spalvas ir brėžimą galima pritaikyti. Piešinys gali būti failas diske, nutolęs URL, įtrauktas į stilių duomenų bazę (*daugiau informacijos*) arba tai gali būti base64 koduota eilutė.

  Simbolis taipogi gali būti nustatytas su *Dinaminiais SVG parametrais*. Apie SVG simbolio parametrizavimą skaitykite skyrių *Parametrizuojami SVG*.

  Pastaba

  SVG versijos reikalavimai

  QGIS braižo SVG failus, kurie atitinka SVG Tiny 1.2 profilį, taikoma įgyvendinimui įvairiuose įrenginiuose, nuo telefonų ir delninukų iki skreitinukų ir stalinių kompiuterių, todėl juose yra tik poaibis savybių iš SVG 1.1 Full, kartu su naujomis savybėmis, išplečiančiomis SVG galimybes.

  Kai kurios savybės, kurios neįtrauktos į šias specifikacijas, QGIS gali būti braižomos neteisingai.

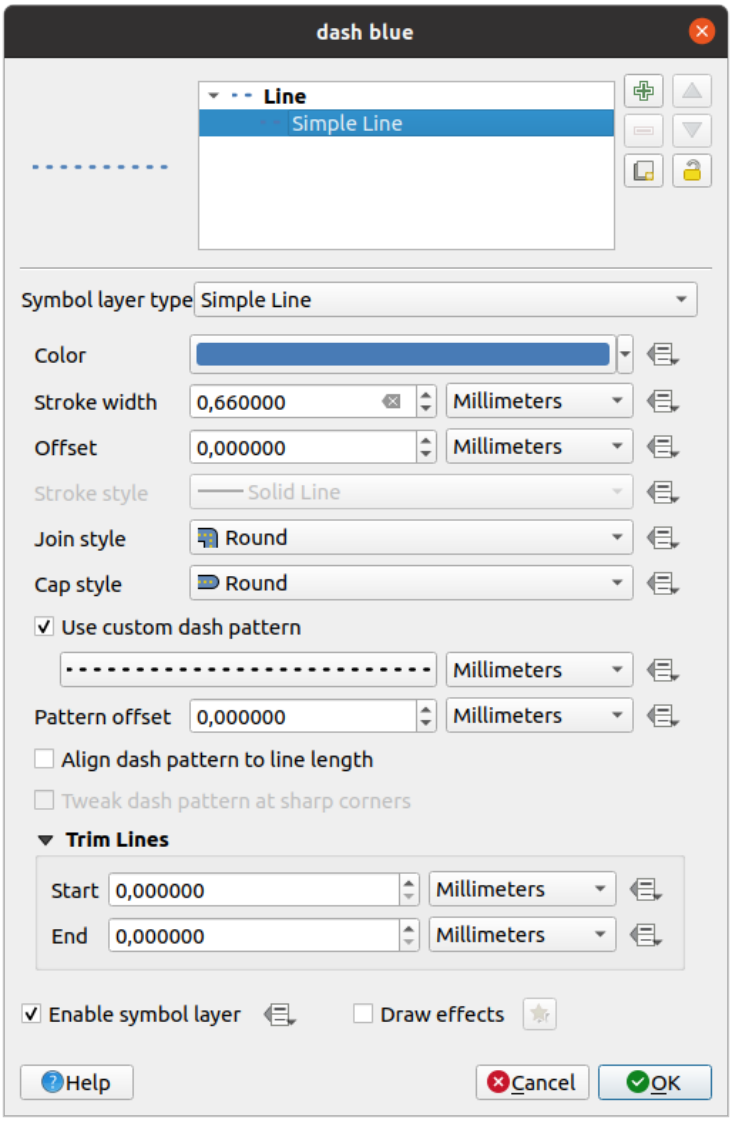
Linijų simboliai

Linijų geometrijų geoobjektams tinkami simboliai turi kelis simbolių sluoksnių tipus:

10.2 Table : List of line symbol layer types

Simple line	*Arrow*	*Filled line*
Geometry generator	*Hashed line*	*Interpolated line*
Linear referencing	*Lineburst*	*Marker line*
Raster line		

- **Paprasta linija** (numatytasis)



10.12 Fig. : Paprasto linijos simbolio kūrimas

Paprastas linijų simbolių sluoksnių tipas turi daug tų pačių savybių kaip ir *paprastas žymeklio simbolis* ir kelis papildomus:

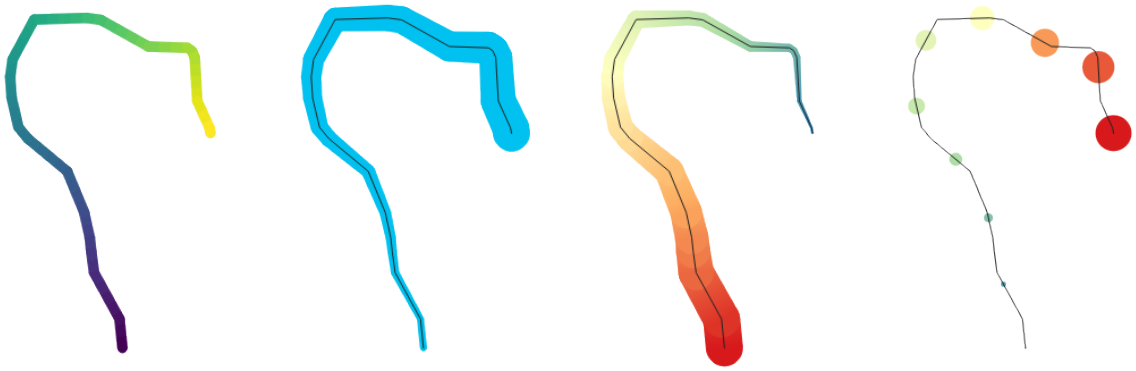
- *Naudoti savo punktyro šabloną*: permuša *Brėžimo stiliaus* nustatymą savo punktyru. Jums reikės nurodyti vieną po kitos einančių brūkšnelių ir tarpų ilgį pasirinktais vienetais, taip sukuriant modelį. Bendras šablono ilgis rodomas dialogo apačioje.

  - *Šablono poslinkis*: brūkšnių/tarpų vieta linijoje gali būti tinkinama taip, kad jie būtų braižomi gražesnėse vietose, atsižvelgiant į linijos kampus (taipogi galima naudoti potencialiai „lygiuoti“ greta einančias brūkšnių šablonų ribas)
  - *Lygiuoti brūkšnių šabloną pagal linijos ilgį*: brūkšnių šablonų ilgis bus pritaikomas taip, kad linija baigtųsi pilnu brūkšnio elementu, o ne tarpu.
  - *Pataisyti punktyro šabloną aštriuose kampuose*: dinamiškai pritaiko punktyrų šablono taikymą taip, kad aštrūs kampai išreiškiami pilnu punktyro elementu einančiu į ir iš aštraus kampo. Priklauso nuo *Pritaikyti punktyrų šabloną pagal linijos ilgį*.
  - *Patrumpinti linijas* nuo *Pradžios* ir/ar *Pabaigos*: leidžia linijos braižymui nukirpti pirmus x mm ir paskutinius y mm nuo realios linijos. Leidžia įvairius *vienetus*, įskaitant procentus nuo bendro linijos ilgio ir gali priklausyti nuo duomenų papildomam valdymui. Pradžios/pabaigos nukirpimo atstumą galima naudoti, pavyzdžiui, kuriant sudėtingus simbolius, kur linijos sluoksnis neturėtų dengti žymeklių simbolių, kurie padėti linijos pradžioje ir pabaigoje.

- **Rodyklė**: braižo linijas kaip kreivių (arba ne) rodykles su viengubomis arba dvigubomis galvomis bei konfigūruojamais (arba nuo duomenų priklausančiais):
  - *Galvutės tipu*
  - *Rodyklės tipu*
  - *Rodyklės pločiu*
  - *Rodyklės pločiu pradžioje*
  - *Galvutės ilgiu*
  - *Galvutės storiu*
  - *Poslinkiu*

  Galima kurti *Kreivines rodykles* (linijos geoobjektas turi turėti bent tris viršūnes) ir *Kartoti rodyklę kiekviename segmente*. Taipogi naudojamas *pildymo simbolis*, kad rodyklės vidus būtų užpildytas gradientu arba pildymu vidun. Kartu su geometrijos generatoriumi, šis sluoksnio simbolio tipas leidžia vaizduoti srauto žemėlapius.

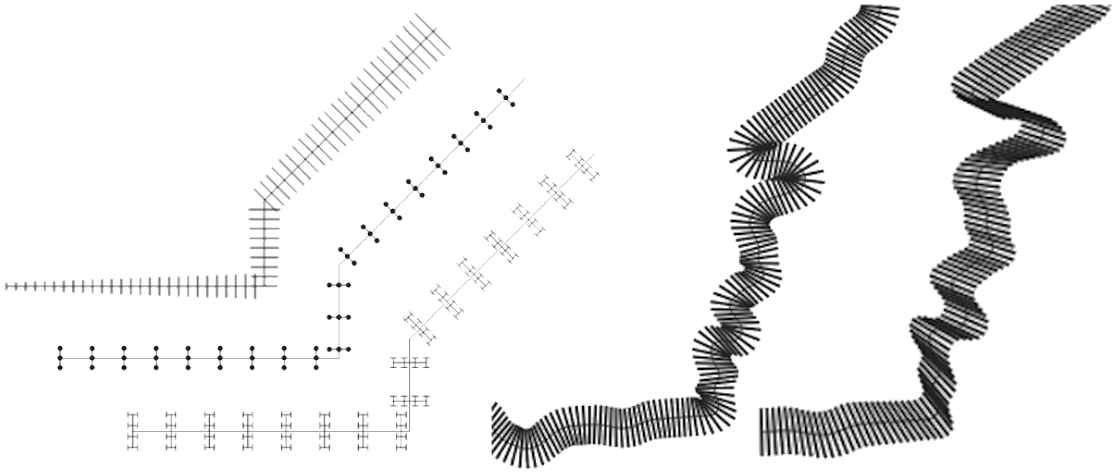
- **Geometrijos generatorius** (žr. *Geometrijos generatorius*)
- **Interpoliuota linija**: leidžia braižyti liniją, kurios *Brėžimo storis* ir/ar *Spalva* gali būti pastovi (nurodyti *Fiksuoto pločio* ir *Vienos spalvos* parametrai) arba gali kisti palei geometriją. Kai kinta, reikalingos įvestys yra:
  - *Pradžios reikšmė* ir *Pabaigos reikšmė*: Reikšmės, kurios bus naudojamos geoobjekto geometrijos interpoliacijos ekstremumuose. Tai gali būti fiksuotos reikšmės, geoobjekto atributai arba priklausyti nuo išraiškos.
  - *Min. reikšmė* ir *Maks. reikšmė*: Reikšmės, tarp kurių vykdoma interpoliacija. Spauskite mygtuką Įkelti, kad automatiškai juos užpildytumėte pagal minimalią ir maksimalią pradžios/pabaigos reikšmes taikomas sluoksniui.
  - Veikia tik brėžimo parinkčiai:
    * *Min. plotis* ir *Maks. plotis*: nurodo intervalą kintančiam pločiui. *Min. pločiui* priskiriama *Min. reikšmė*, o *Maks. pločiui* - *Maks. reikšmė*. Gali būti nurodytas *vienetas*.
    * *Naudoti absoliučią reikšmę*: interpoliacijai naudoti tik absoliučią reikšmę (neigiamos reikšmės naudojamos kaip teigiamos).
    * *Ignoruoti už ribų*: pagal nutylėjimą, jei geoobjekto intervalas `[pradinė reikšmė – galutinė reikšmė]` nepatenka į intervalą `[min. reikšmė – maks. reikšmė]`, iš ribų išeinančios geoobjekto geometrijos dalys braižomos minimaliu ar maksimaliu pločiu. Įjunkite šią parinktį, kad jos išvis nebūtų braižomos.
  - Kintamai spalvai galite naudoti bet kurį iš *spalvų rampos klasifikacijos* interpoliavimo metodų



10.13 Fig. : Interpoliuotų linijų pavyzdžiai

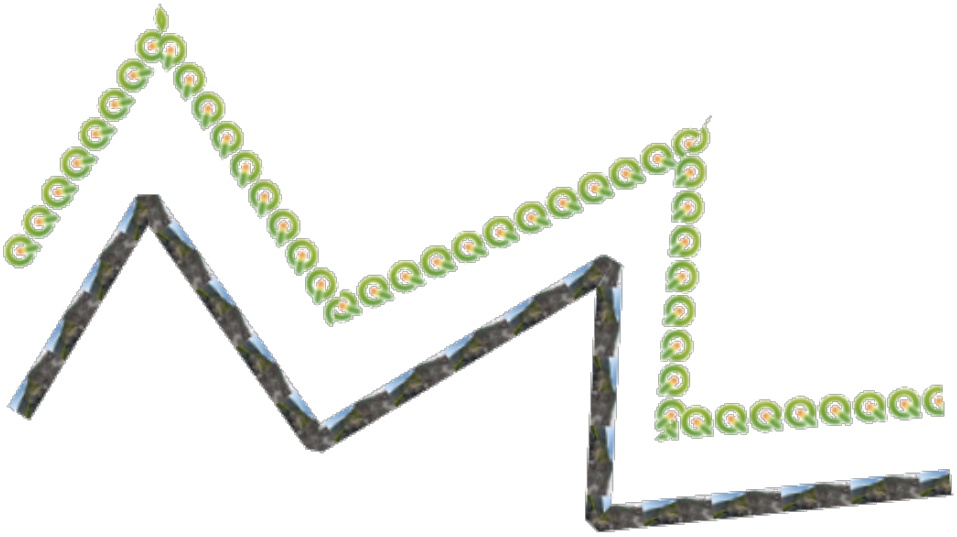
- **Žymeklių linija**: kartoja *žymeklio simbolį* per visą linijos ilgį.
  - *Žymeklio vietą* galima nurodyti naudojant paprastą intervalo nustatymą (pradedant nuo pirmos viršūnės) ir/arba linijos geometrijos savybę (pirmoje ar paskutinėje viršūnėje, vidinėse viršūnėse, linijos ar kiekvieno segmento centre arba kiekviename kreivės taške).
  - Kai įjungta pirmos ar paskutinės viršūnės pozicija, parinktis *Padėti kiekvienoje ekstremumo dalyje* nurodys, kad žymeklius reikia braižyti ir pirmoje ar paskutinėje kelių dalių geometrijų dalyse.
  - *Poslinkis palei liniją*: žymeklių pozicijai taipogi galima nurodyti poslinkį palei liniją jūsų pasirinktais ref:*vienetais* <*unit_selector*> (milimetrais, taškais, žemėlapio vienetais, metrais mastelyje, procentais…):
    - Teigiama reikšmė paslenka žymeklių simbolius linijos kryptimi (su pozicijomis *Ant pirmos viršūnės* ir *Su intervalu*) ir priešinga kryptimi (su pozicija *Ant paskutinės viršūnės*).
    - Neigiama reikšmė su neuždara linija reikš jokio poslinkio (pozicijoms *Ant pirmos viršūnės* ir *Ant paskutinės viršūnės*) arba atvirkštinį simbolių poslinkį (nuo paskutinės viršūnės).
    - Uždaro žiedo atveju QGIS poslinkį laiko kaip besitęsiantį aplink žiedą (pirmyn arba atgal). Pvz. nustačius poslinkį 150% (atitinkamai -10% ar -110%), bus gautas 50% poslinkis (atitinkamai 90%) uždaro žiedo ilgio.
  - Parinktis *Sukti žymeklį pagal linijos kryptį* nurodo, ar reikia kiekvieną žymeklio simbolį pasukti pagal linijos kryptį.

    Kadangi linija dažnai yra seka skirtingų krypčių segmentų, žymeklio pasukimas skaičiuojamas kaip nurodyto ilgio palei liniją vidurkis. Pavyzdžiui nurodžius savybės *Vidutinis kampas per* reikšmę `4mm`, bus paimti du taškai linijoje, esantys `2mm` prieš ir po simbolio vietos ir taip bus paskaičiuotas žymeklio simbolio pasukimas. Taip glotninamas (arba pašalinamas) bet koks mažas vietinis nuokrypis nuo bendros linijos krypties, tokiu būdu gaunant daug gražesnę vizualią linijų žymeklių simbolių kryptį.
  - *Linijos poslinkis*: žymeklių simbolius taipogi galima paslinkti nuo linijos geoobjekto.
- **Brūkšnių linija**: kartoja linijos segmentą (brūkšnį) per linijos ilgį su linijos sub-simboliu, kuris naudojamas kiekvieno individualaus segmento braižymui. Kitais žodžiais, brūkšnių linija yra kaip žymeklių linija, kurios žymeklių simboliai pakeisti segmentais. Kaip tokios, brūkšnių linijos turi *tas pačias savybes* kaip žymeklių simboliai, kartu su:
  - *Brūkšnio ilgis*
  - *Brūkšnio pasukimas*



10.14 Fig. : Brūkšnių linijų pavyzdžiai

- **Rastro linija**: braižo ir kartoja rastro piešinį palei linijos geoobjekto formą. Galima keisti *Brėžimo plotį*, *Poslinkį*, *Jungimo stilių*, *Pabaigos stilių* ir *Nepermatomumą*.



10.15 Fig. : Rastro linijų pavyzdžiai

- **Tiesinė atskaita**: leidžia teksto užrašus dėlioti pastoviais intervalais palei liniją arba esamų viršūnių pozicijose. Pozicijas galima skaičiuoti naudojant dekarto atstumus arba interpoliuojant pagal z/m reikšmes. Tarp parametrų yra:
  - *Matavimo pozicija*: Užrašus galima dėlioti pastoviais 2d dekarto atstumais, paprastai tiesiškai interpoliuojant tarpus naudojant geometrijų Z ar M reikšmes arba esamose viršūnėse. Nurodomas fiksuotas arba nuo duomenų priklausantis *Intervalas*.
  - *Kiekybė*: Kiekvienoje pozicijoje rodomi užrašai reprezentuoja einamąjį atstumą linijoje arba tiesiškai interpoliuotą Z ar M reikšmę. Užrašai braižomi naudojant visą diapazoną *teksto* ir *skaičių* formatavimo funkcionalumo, kurį teikia QGIS.

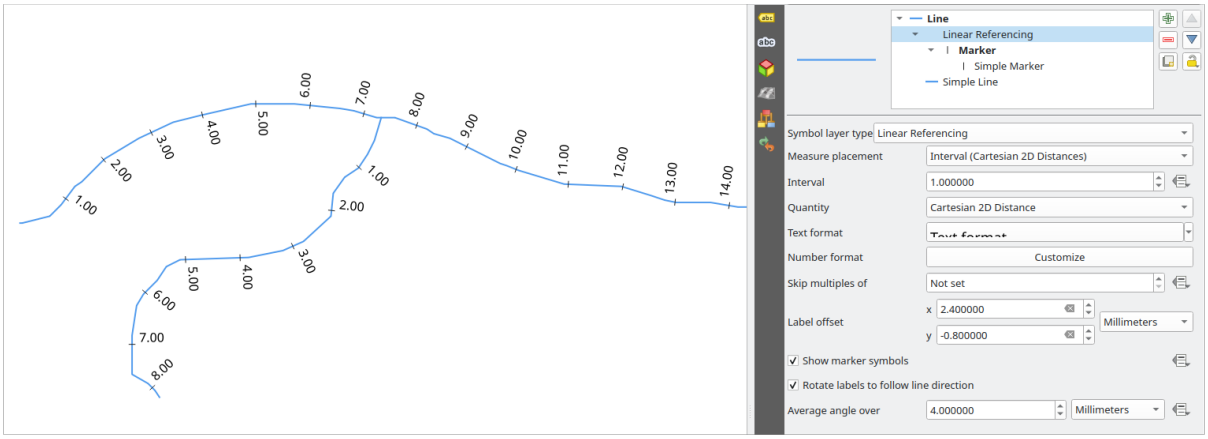
Pastaba

Kai naudojamas atstumu paremtas užrašų dėliojimas, atstumai skaičiuojami naudojant tik 2D. Dekarto skaičiavimai remiasi originalaus sluoksnio CRS.

– *Praleisti daugiklius*: Jei nustatyta, užrašai, kurie yra šios reikšmės daugikliai, bus praleisti. Tai leidžia konstruoti sudėtingus atskaitos užrašus, pavyzdžiui kur simbolis turi du tiesinės atskaitos simbolių sluoksnius, kur vieni užrašai dėliojami kas 100m mažu šriftu, praleidžiant 1000, o antras sluoksnis dėlioja užrašus kas 1000m dideliu šriftu.

– *Kampo vidurkis per*: Užrašai braižomi naudojant kampą, paskaičiuotą naudojant linijos vidurkį, taigi užrašai nebus stipriai išlankstyti dėl mažų aštrių kampų.

– *Rodyti žymeklio simbolius*, nurodytuose linijos geoobjekto taškuose naudojamas pilnas QGIS žymeklio simbolis. Tai leidžia, pavyzdžiui, rodyti perbraukimą taške su užrašu, taip sukuriant „liniuotės“ stiliaus liniją.

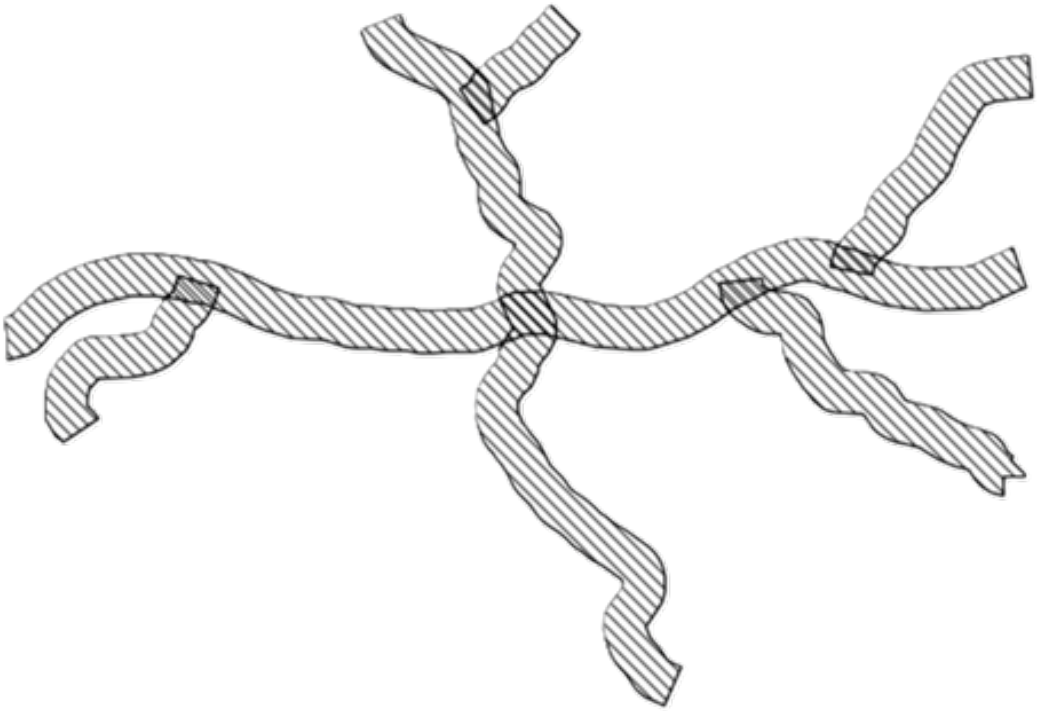


10.16 Fig. : Linijos užrašų braižymas palei linijos geoobjektą

- **Linijų srautas**: braižo gradientą palei liniją. Galite parinkti *Dvi spalvas* arba *Spalvų rampą* ir *Brėžimo plotį*, *Poslinkį*, *Jungimo stilių*, *Pabaigos stilių*.

10.17 Fig. : Linijų srauto pavyzdžiai

- **Užpildyta linija**: braižo linijų vidų naudojant pildymo simbolį taip leidžiant kurti linijas, užpildytas gradientais, linijų punktyrais ir pan. Galima keisti *Braižymo plotį*, *Poslinkį*, *Jungimo stilių*, *Pabaigos stilių*.



10.18 Fig. : Užpildytų linijų pavyzdžiai

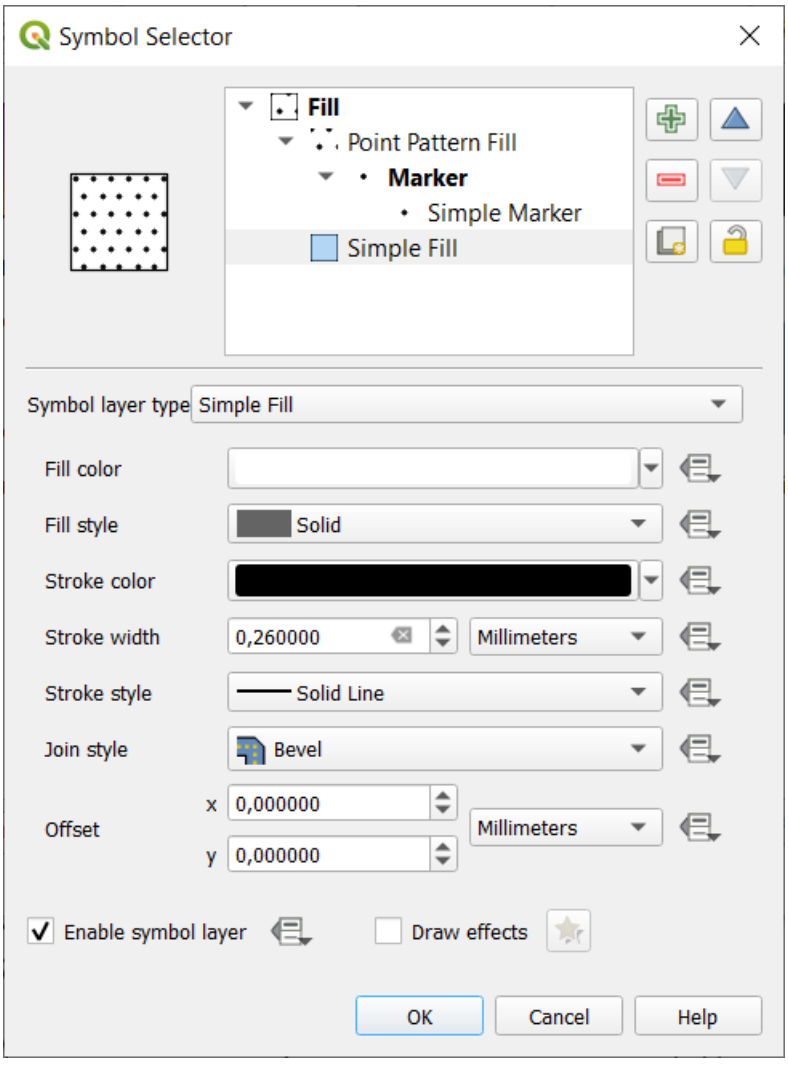
Pildymo simboliai

Poligonų geometrijų geoobjektams tinkami pildymo simboliai taipogi turi keletą simbolinių sluoksnių tipų:

10.3 Table : List of fill symbol layer types

For polygon interior	For polygon ring
Simple fill	*Outline: Arrow*
Centroid fill	*Outline: Filled line*
Geometry generator	*Outline: Hashed line*
Gradient fill	*Outline: Interpolated line*
Line pattern fill	*Outline: Linear referencing*
Point pattern fill line	*Outline: Lineburst*
Random marker fill	*Outline: Marker line*
Raster image	*Outline: Raster line*
SVG fill	*Outline: Simple line*
Shapeburst fill	

- **Paprastas pildymas** (numatytasis): pildo poligoną viena spalva



10.19 Fig. : Paprasto pildymo simbolio kūrimas

- **Centroido pildymas**: padeda *žymeklio simbolį* į matomo geoobjekto centrą. Žymeklio pozicija negali būti

realus geoobjekto centroidas, nes skaičiavimas atsižvelgia tik į poligoną(us), kurie apkarpyti pagal matomą žemėlapio plotą braižymui ir ignoruoja skyles. Jei norite tikslaus centroido, naudokite *geometrijos generatoriaus simbolį*.

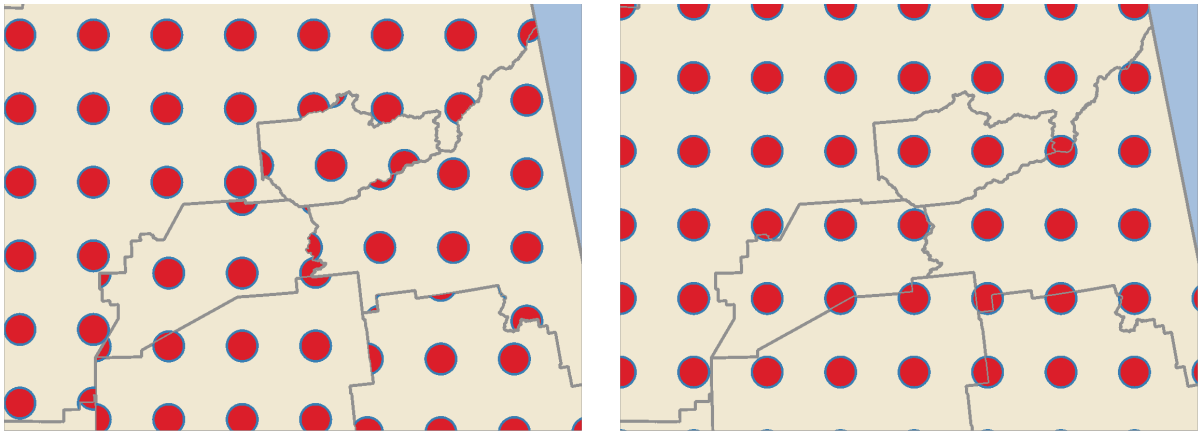
Jūs galite:

- *Priversti žymeklių dėliojimą poligonų viduje*
- *Braižyti žymeklius kiekvienoje kelių dalių geoobjektų dalyje* arba brėžti tašką tik didžiausioje dalyje
- rodyti žymeklio simbolį(ius) pilnus ar dalinius, paliekant tas dalis, kurios išeina už dabartinio geoobjekto geometrijos (*Apkirpti žymeklius pagal poligono geometriją*) ar geometrijos dalį, kuriai priklauso simbolis (*Apkarpyti žymeklius tik pagal dabartinės dalies ribą*)

- **Geometrijos generatorius** (žr. *Geometrijos generatorius*)
- **Gradiento užpildymas**: naudoja spindulinį, tiesinį ar kūginį gradientą, paremtą arba paprastu dviejų spalvų gradientu, arba iš anksto apibrėžta *gradiento spalvų rampa* poligonų pildymui. Gradientą galima pasukti ir taikyti kiekvienam geoobjektui atskirai arba bendrai visai žemėlapio apimčiai. Taipogi galima nurodyti pradžios ir pabaigos taškus tiek koordinatėmis, tiek ir naudojant centroidą (geoobjekto ar žemėlapio). Galima apibrėžti nuo duomenų priklausantį poslinkį.
- **Linijos šablono pildymas**: pildo poligoną *linijų simbolio sluoksnio* štrichavimu. Jūs galite nurodyti:
  - *Lygiavimas*: apibrėžia kokia šablono vieta santykinai nuo geoobjekto(ų):
    - *Lygiuoti šabloną su geoobjektu*: linijos braižomos kiekviename geoobjekte
    - *Lygiuoti šabloną pagal žemėlapio apimtį*: šablonas braižomas visoje žemėlapio apimtyje, taip leidžiant, kad šablonas gražiai susiderintų su visais geoobjektais
  - *Pasukimas* linijų, prieš laikrodžio rodyklę
  - *Tarpas*: atstumas tarp viena po kitos einančių linijų
  - *Poslinkis* linijų atstumas nuo geoobjektų ribos
  - *Apkirpimas*: leidžia valdyti, kaip linijos užpildyme apkarpomos pagal poligono formą. Parinktys yra:
    - *Apkarpyti tik braižymo metu*: kuriamos visą geoobjekto apimantį stačiakampį dengiančios linijos ir tada jos apkerpamos braižymo metu. Linijų ekstremumai (pradžioje ir pabaigoje) nebus matomi.
    - *Apkirpti linijas prieš braižymą*: linijos apkerpamos pagal tikslią poligono formą prieš braižymą. Linijų ekstremumai (įskaitant pabaigos stilius, pradžios/pabaigos žymeklių linijų objektus, …) bus matomi ir kartais gali išeiti už poligono (priklausomai nuo linijų simbolių nustatymų).
    - *Be apkarpymo*: nedaromas joks apkarpymas - linijos dengs visą geoobjektą apimantį stačiakampį

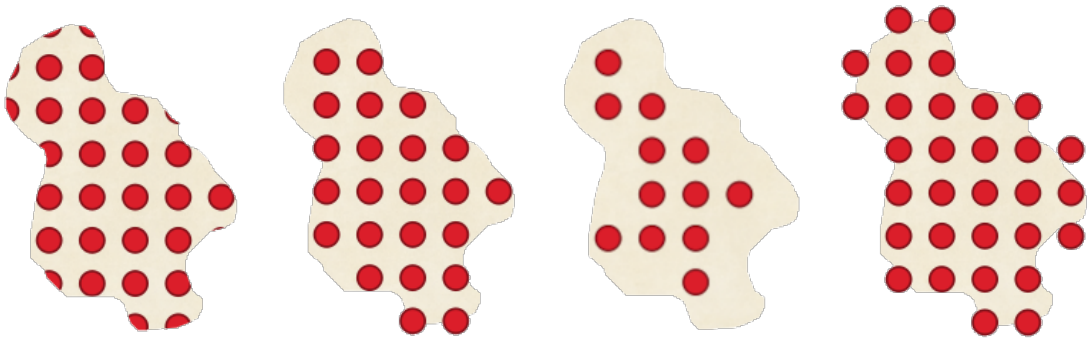
> **Pastaba**
>
> When exporting maps to vector formats such as PDF or SVG, or when the line symbol in the pattern has data-defined properties, QGIS renders the line pattern fill using a line-by-line vector-based approach. This results in smaller output file sizes, higher quality exports, and improved editability of the pattern lines in external vector graphics applications.

- **Taškų šablono pildymas**: pildo poligoną tinklelio šablonu, sudarytu iš *žymeklio simbolių*. Jūs galite nurodyti:
  - *Lygiavimas*: apibrėžia kokia šablono vieta santykinai nuo geoobjekto(ų):
    - *Lygiuoti šabloną pagal geoobjektą*: žymeklių linijos braižomos kiekvieno geoobjekto viduje
    - *Lygiuoti šabloną pagal žemėlapio apimtį*: šablonas braižomas visoje žemėlapio apimtyje, taip leidžiant, kad žymekliai gražiai susiderintų su visais geoobjektais



10.20 Fig. : Taškų šablono lygiavimas pagal geoobjektą (kairėje) ir žemėlapio apimtį (dešinėje)

- *Atstumas*: *Horizontalus* ir *Vertikalus* atstumas tarp vienas po kito einančių žymeklių
- *Poslinkis*: a *Horizontalus* (atit. *Vertikalus*) lygiavimo poslinkis tarp vienas po kito einančių žymeklių stulpelyje (atit. eilutėje)
- *Poslinkis*: *Horizontalus* ir *Vertikalus* atstumas nuo geoobjekto ribos
- *Apkirpimas*: leidžia valdyti, kaip žymekliai užpildyme apkarpomi pagal poligono formą. Parinktys yra:
  - *Apkirpti pagal formą*: žymekliai apkerpami taip, kad matomos būtų tik tos dalys, kurios yra poligono viduje
  - *Žymeklio centroidas formoje*: braižomi tik tie žymekliai, kurių centras patenka į poligono vidų, bet pačių žymeklių dalys už poligono ribų nėra apkerpamos
  - *Žymeklis pilnai formoje*: rodomi tik tie žymekliai, kurie pilnai patenka į poligoną
  - *Be apkirpimo*: bet kokie žymekliai, kurie kertasi su poligonu bus pilnai braižomi (griežtai sakant „kertasi su žymeklio apimties stačiakampiu“)



10.21 Fig. : Žymeklių apkarpymas užpildyme - iš kairės į dešinę: apkarpyti pagal formą, žymeklio centroidas formoje, žymeklis pilnai formoje, be apkarpymo

- *Pasukimas* viso šablono, pagal laikrodžio rodyklę
- The *Atsitiktinio šablono* grupinis nustatymas leidžia kiekvieną šablono užpildymo tašką šiek tiek atsitiktinai pastumti iki nurodyto maksimalaus atstumo *Horizontaliai* ar *Vertikaliai*. Jūs galite nurodyti maksimalų poslinkį bet kokiais palaikomais vienetais, tokiais kaip milimetrai, taškai, žemėlapio vienetai ar net „procentai“ (kur procentai bus santykis su šablono pločiu ar aukščiu).

  Jūs galite nurodyti neprivalomą atsitiktinių skaičių sėklą, jei norite išvengti simbolių šablonų „šokinėjimo“ atnaujinant žemėlapį. Taipogi palaikomi ir nuo duomenų priklausantys permušimai.

> **Pastaba**
>
> Pagrindinis skirtumas tarp *Atsitiktinio šablono* ir *atsitiktinio žymeklių pildymo* simbolių tipų yra tai, kad atsitiktinis poslinkis su taškų šablonu leidžia pusiau-„reguliarų“ žymeklių išdėstymą - nes taškai šablone efektyviai apriboti gardele, tai leidžia kurti pusiau atsitiktinius pildymus, kurie neturi tuščių plotų ar persidengiančių žymeklių. (Priešingai nei atsitiktinių žymeklių pildymas, kuris visada taškus išdėstys visiškai atsitiktinai… kartais gaunant vizualius spiečius ar nenorimus tuščius plotus).

- **Atsitiktinių žymeklių užpildymas**: pildo poligoną *žymeklių simboliais* atsitiktinai išmėtytais poligono ribose. Jūs galite nurodyti:
  - *Skaičiavimo būdą*: ar braižomų žymeklių simbolių skaičius nurodomas kaip absoliutus skaičius, ar remiasi tankiu
  - *Taškų skaičius*: braižomų žymeklių simbolių skaičius,
  - neprivalomas atsitiktinis skaičius *sėkla*, kad būtų pastovus išdėstymas
  - *Tankio plotas*: jei naudojamas tankiu paremtas skaičiavimo metodas, užtikrina, kad žymeklių užpildymo tankis bus toks pat skirtinguose masteliuose kai žemėlapiai atnaujinami (taipogi leidžia atsitiktiniam išdėstymui gražiai veikti su QGIS serveriu ir nuo laiko priklausančiu braižymu)
  - *Apkarpyti žymeklius pagal poligono ribą*: ar žymeklius, braižomus šalia poligonų ribų, reikia apkarpyti pagal poligono ribą ar ne
- **Rastro piešinio pildymas**: pildo poligoną rastro piešinio kaladėlėmis (`PNG` `JPG`, `BMP` …). Piešinys gali būti failas diske, nutolęs URL ar įdėtas failas, užkoduotas kaip eilutė (*daugiau informacijos*). Galimos (nuo duomenų priklausančios) parinktys yra nepermatomumas, dydis, koordinačių režimas (objektas ar vaizdas), pasukimas ir poslinkis. Su dydžio parinktimi jūs galite nepriklausomai keisti pildymo šablono plotį ir aukštį, taip įgalinant ištemptus rastro pildymus tiek horizontalia, tiek vertikalia kryptimi. Piešinio plotį ir aukštį galima nurodyti naudojant *bendrus vienetus* arba procentais nuo pradinio dydžio.
- **SVG pildymas**: pildo poligoną nurodyto dydžio (*Tekstūros pločio*) *SVG žymekliais*.
- **Išdeginimo pildymas**: gradiento pildymo buferis, kur gradientas paišomas nuo poligono ribos link poligono centro. Tarp konfigūruojamų parametrų yra spalvinamas atstumas nuo ribos, spalvos rampų naudojimas ar paprastas dviejų spalvų gradientas, pasirinktinai užpildymo ir kraštų išblukinimas.
- **Outline: Arrow**: uses a line *arrow symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The settings for the outline arrow are the same as for arrow line symbols.
- **Outline: Filled line**: uses a *filled line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The settings are the same as for filled line symbols.
- **Outline: Hashed line**: uses a *hash line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for hashed line symbols.
- **Outline: Interpolated line**: uses an *interpolated line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for interpolated line symbols.
- **Outline: Linear referencing**: uses a *linear referencing symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for linear referencing symbols.
- **Outline: Lineburst**: uses a *lineburst symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for lineburst symbols.
- **Outline: Marker line**: uses a *marker line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings for the outline marker line are the same as for marker line symbols.

- **Outline: Raster line**: uses a *raster line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings for the outline hashed line are the same as for hashed line symbols.
- **Outline: Simple line**: uses a *simple line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The *Draw line only inside polygon* option displays the polygon borders inside the polygon and can be useful to clearly represent adjacent polygon boundaries. The other settings for the outline simple line are the same as for simple line symbols.

Pastaba

Kai geometrijos tipas yra poligonas, jūs galite pasirinkti išjungti automatinį linijų/poligonų apkarpymą pagal drobės apimtį. Kai kuriais atvejais dėl apkarpymo gaunama netinkama simbologija (pvz. centroido užpildymai, kur centroidas visada turi būti tikras geoobjekto centroidas).

Parametrizuojami SVG

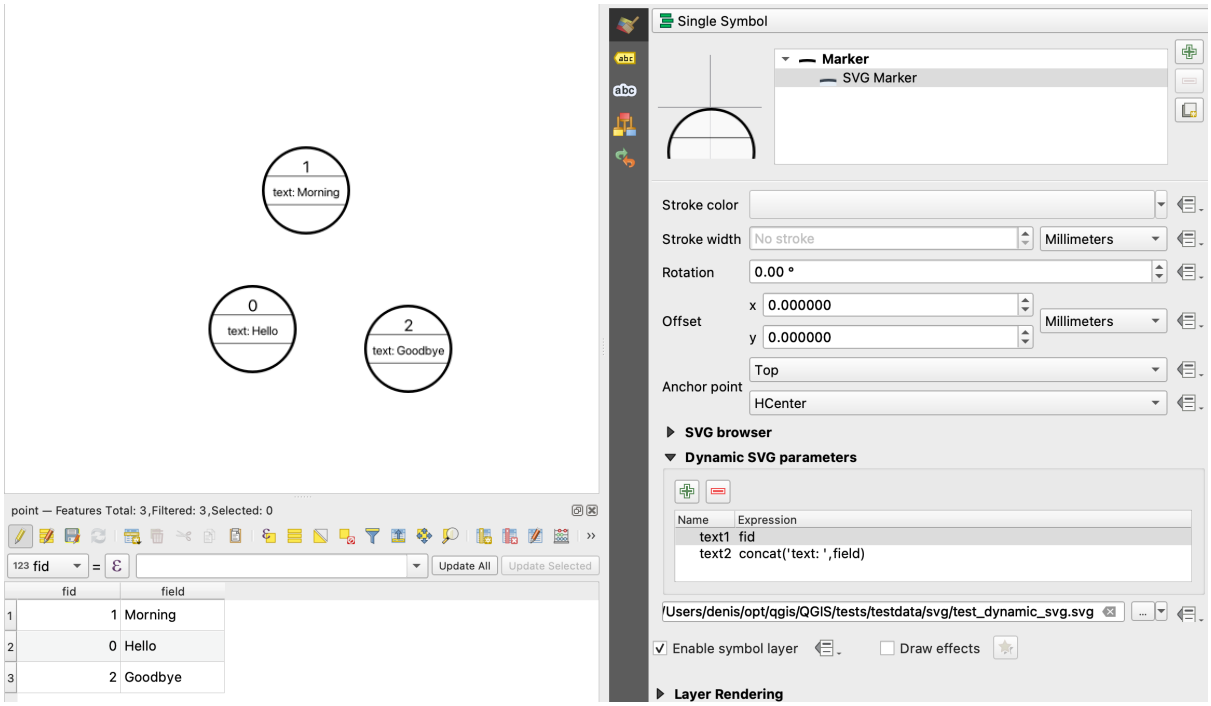
Jūs turite galimybę pakeisti *SVG žymeklio* spalvas. Turite pridėti parametrus `param(fill)` pildymo spalvai, `param(fill-opacity)` pildymo nepermatomumui, `param(outline)` ir `param(outline-opacity)` brėžimo spalvai ir nepermatomumui atitinkamai, bei `param(outline-width)` brėžimo pločiui. Po šių parametrų gali sekti numatytoji reikšmė, pvz.:

```
<svg width="100%" height="100%">
<rect fill="param(fill) #ff0000" fill-opacity="param(fill-opacity) 1" stroke=
↪"param(outline) #00ff00" stroke-opacity="param(outline-opacity) 1" stroke-width=
↪"param(outline-width) 10" width="100" height="100">
</rect>
</svg>
```

Bendrai paėmus SVG galima laisvai parametrizuoti naudojant `param(parametro_pavadinimas)`. Šį parametrą galima naudoti arba kaip atributo reikšmę, arba mazgo tekstą:

```
<g stroke-width=".265" text-anchor="middle" alignment-baseline="param(align)">
  <text x="98" y="147.5" font-size="6px">param(text1)</text>
  <text x="98" y="156.3" font-size="4.5px">param(text2)</text>
</g>
```

Tada parametrus galima apibrėžti kaip išraiškas lentelėje *Dinaminiai SVG parametrai*.



10.22 Fig. : Dinaminių SVG parametrų lentelė

Pastaba

QGIS ieško pilno SVG mazgo. Taigi, jei jūsų parametras yra sudėtingesnio mazgo viduje, jums reikia įterpti į išraišką pilną mazgą. Pavyzdžiui `transform="rotate(param(angle)"` neveiks. Vietoje to, jums reikia daryti `transform="param(rotation)"` ir `rotation` parametras bus apibrėžtas išraiškoje `'rotate(' || coalesce(my_field, 0) || ')'`.

Geometrijos generatorius

Su visais simbolių tipais veikiantis *geometrijos generatoriaus* simbolių sluoksnis leidžia naudoti *išraiškų sintaksę* dinamiškai kuriant geometrijas braižymo metu. Gauta geometrija neturi atitikti pradinį *Geometrijos tipą* ir jūs galite pridėti kelis skirtingai pakeistus simbolinius sluoksnius vieną ant kito.

Galima nurodyti savybę *Vienetai*: kai geometrijos generatoriaus simbolis nėra taikomas sluoksniui (t.y. kai jis naudojamas išdėstymo elementui), tai leidžia geriau valdyti sukurtą išvestį.

Keli pavyzdžiai:

- Braižyti simbolį kaip geoobjekto centroidą

```
centroid( $geometry )
```

- Vizualiai sukirsti geoobjektus 100 žemėlapio vienetų atstumu nuo taško geoobjekto, t.y. kurti 100m buferį aplink tašką

```
buffer( $geometry, 100 )
```

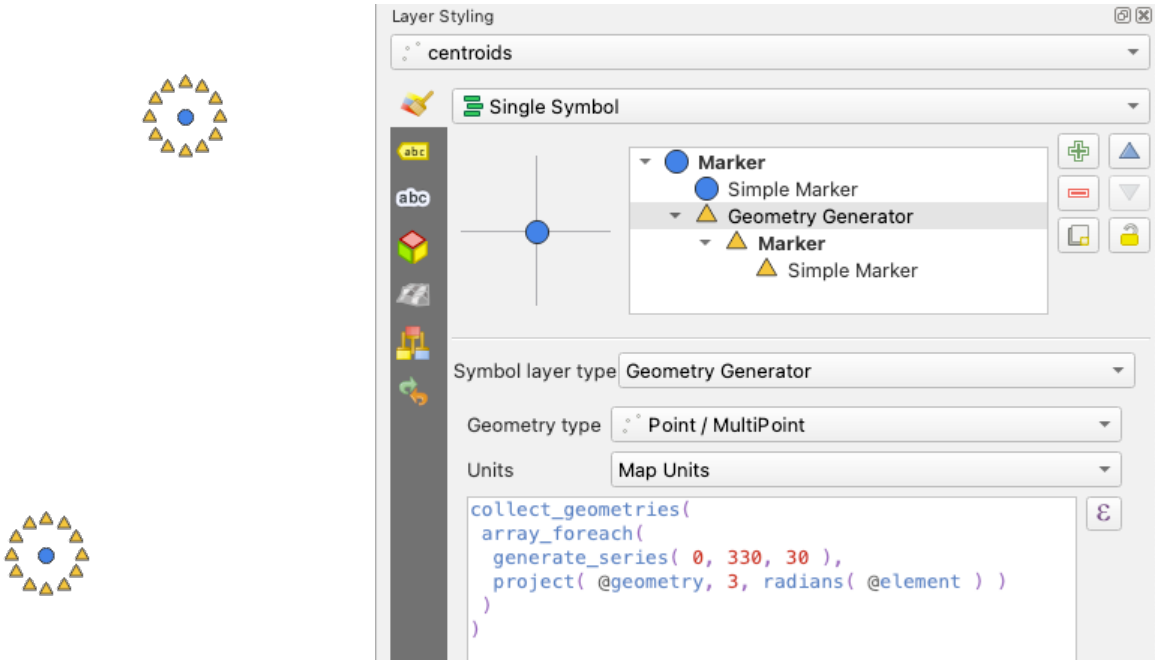
- Sukurti ratu išsidėsčiusius taškus aplink centrinį geoobjekto tašką, naudojant juos kaip MultiPoint geometrijos generatorių

```
collect_geometries(
  array_foreach(
```

(continues on next page)

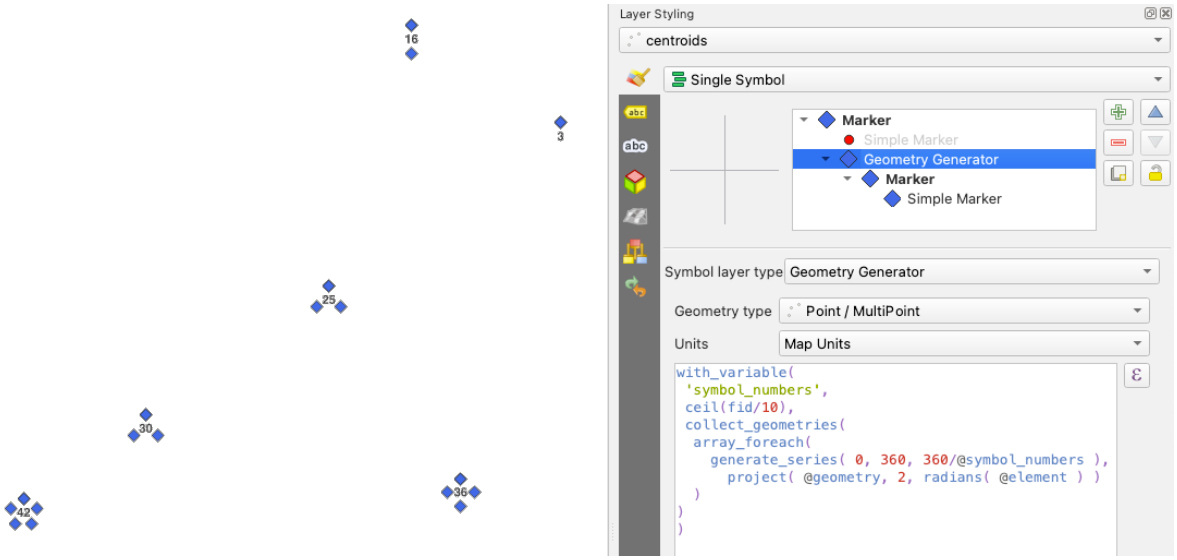
(tęsinys iš praeito puslapio)

```
    generate_series( 0, 330, 30 ),
    project( $geometry, 3, radians( @element ) )
  )
)
```



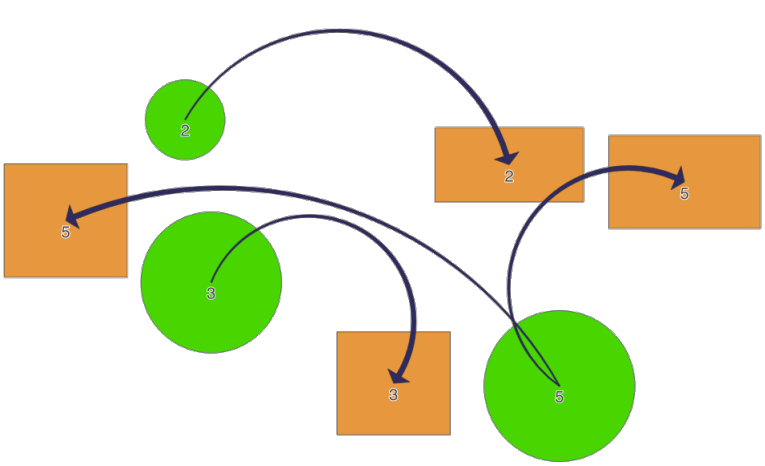
- Kurti ratu išsidėsčiusius taškus aplink centrinį geoobjekto tašką. Taškų skaičiuos priklauso nuo lauko reikšmės.

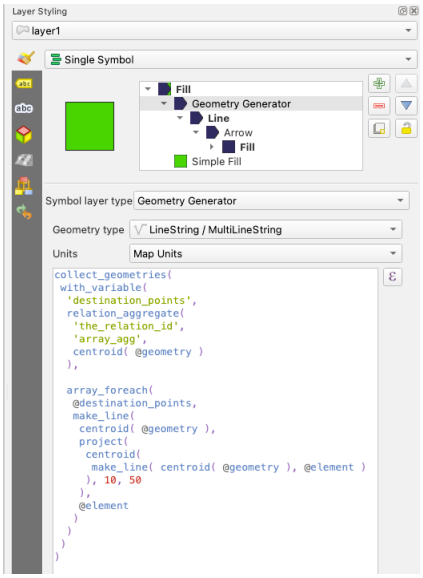
```
with_variable(
  'symbol_numbers',
  ceil(fid/10),
  collect_geometries(
    array_foreach(
      generate_series( 0, 360, 360/@symbol_numbers ),
      project( $geometry, 2, radians( @element ) )
    )
  )
)
```



- Kurti rodyklę turinčia kreivę, jungiančią dviejų sluoksnių geoobjektus naudojant jų *ryšį*

```
collect_geometries(
  with_variable(
    'destination_points',
    relation_aggregate( 'the_relation_id', 'array_agg', centroid( $geometry )␣
↪),
    array_foreach(
      @destination_points,
      make_line(
        centroid( @geometry ),
        project(
          centroid( make_line( centroid( @geometry ), @element ) ),
          10,
          50
        ),
        @element
      )
    )
  )
)
```





Vektorinio lauko žymeklis

Vektorinio lauko žymeklis naudojamas rodant vektorinio lauko duomenis, tokius kaip žemės deformacija, potvynių srautai ir pan. Jis rodo vektorius kaip linijas (paprastai rodykles), kurios padidintos ir orientuotos pagal parinktus duomenų taškų atributus. Jį galima naudoti tik braižant taškų duomenis; linijų ir poligonų sluoksniai su šia simbologija nebraižomi.

Vektorinis laukas apibrėžiamas pagal duomenų atributus, kurie gali reprezentuoti lauką:

- **dekarto** komponentais (lauko `x` ir `y` komponentai)
- **polinės** koordinatėmis: šiuo atveju, atributai apibrėžia `Ilgį` ir `Kampą`. Kampą galima matuoti arba pagal laikrodžio rodyklę nuo šiaurės, arba prieš laikrodžio rodyklę nuo rytų, ir galima išreikšti tiek laipsniais, tiek radianais.
- **tik aukščio** duomenimis, kurie rodo vertikalias rodykles, padidintas pagal duomenų atributą. Tai tinkama, pavyzdžiui, rodant vertikalų deformacijos komponentą.

Lauko stiprumą galima didinti ar mažinti iki reikiamo dydžio lauko peržiūrai.

Animuotas žymeklis

Animuoto žymeklio simbolio tipas leidžia jums naudoti `.GIF`, `.WebP`, `.MNG`, kt. animacijos failą reprezentuojant taškus jūsų žemėlapyje. Jūs galite nurodyti:

- *Failo* kelią,
- *Kadrų skaičius*: skaičius žingsnių, kurie rodomi per sekundę, taip nurodant, kaip greitai grojama animacija,
- *Dydį* bet kuriais *palaikomais vienetais*,
- *Nepermatomumą*,
- *Pasukimą*,
- *Poslinkį x* ir *y* kryptimis nuo žymeklio pozicijos,
- *Pririšimo taškas*

Yra du animuotų simbolių valdymo būdai:

- **Kai jūsų žemėlapis nėra sukonfigūruotas kaip animacija** (t.y. jis yra standartinis QGIS projektas be animacijų), animuotų žymeklių kadras bus nustatomas išskirtinai tik pagal dabartinę laiko žymą.
- **Kai jūsų žemėlapis yra** *sukonfigūruotas kaip animacija*, animuoti žymekliai sinchronizuosis su animacijos laiko juosta. Tai reiškia, kad animuoti žymekliai sustos, kai animacija sustabdoma, judės pirmyn kartu su animacija ir t.t.. Žemėlapis taipogi bus perbraižomas pagal kadrų dažnį, nurodytą laiko animacijai. Šis režimas taipogi taikomas, kai animacija eksportuojama naudojant laiko valdiklį.

10.3 Užrašo nustatymas

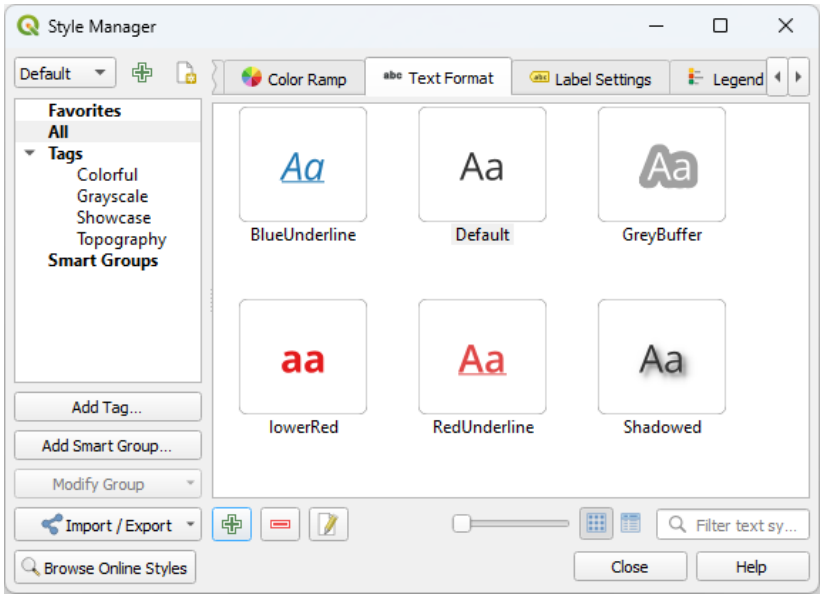
Labels are textual information you can display on maps. They add details you could not necessarily represent using symbols, and may refer to vector features, raster cells, mesh elements, or simple annotations on the map… Two types of text-related items are available in QGIS:

- *Teksto formatas*: apibrėžia teksto išvaizdą, įskaitant *šriftą, dydį, spalvas*, *šešėlį*, *foną*, *buferį*, …

  Juos galima naudoti braižant tekstą virš žemėlapio (išdėstymo/žemėlapio pavadinimas, dekoracijos, mastelio juosta, …), paprastai naudojant valdiklį *šriftas*.

  Norėdami sukurti *Teksto formato* elementą:

  1. Atverkite dialogą *Stilių tvarkyklė*
  2. Įjunkite kortelę *Teksto formatas*



10.23 Fig. : Teksto formatai Stilių tvarkyklės dialoge

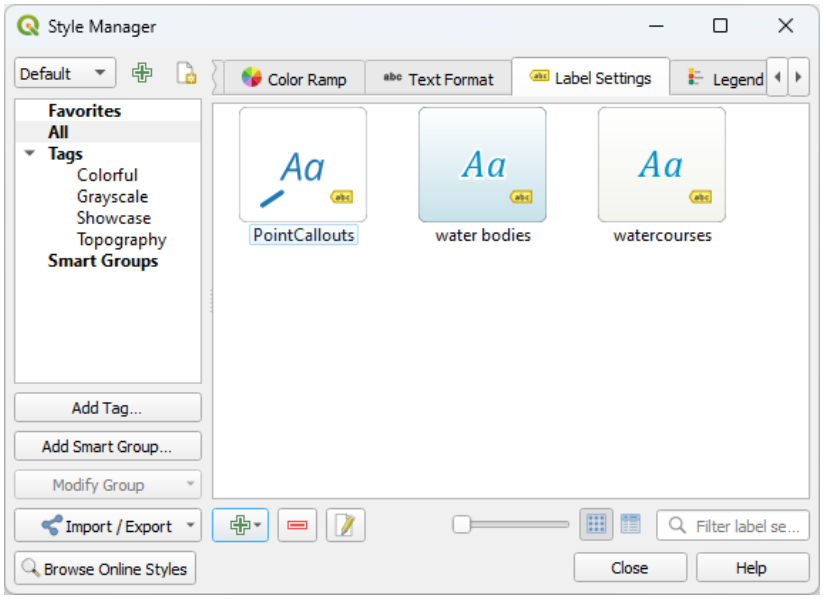
3. Spauskite mygtuką Pridėti elementą. Atveriamas dialogas *Teksto formatas* su *konfigūracija*. Kaip paprastai, šias savybes galima *apibrėžti priklausomai nuo duomenų*.

- *Label Settings*: extend the text format settings with properties related to the location or the interaction with other texts or features (*callouts*, *placement*, *overlay, scale visibility*, *mask* …).

  Jie naudojami konfigūruojant išmanius vektorinių ir tinklelio sluoksnių užrašus naudojant kortelę *Užrašai*, esančią vektorinių ir tinklo *Sluoksnio savybių* dialoge, ar *Sluoksnio stiliaus* skydelį, ar naudojant mygtuką Sluoksnio užrašų parinktys, esantį *Užrašų įrankinėje*.

  Norėdami sukurti *Užrašų nustatymų* elementą:

  1. Atverkite dialogą *Stilių tvarkyklė*
  2. Įjunkite kortelę *Užrašų nustatymai*



10.24 Fig. : Užrašų nustatymai stiliaus tvarkyklės dialoge

3. Spauskite mygtuką Pridėti elementą ir parinkite įrašą, atitinkantį geometrijos tipą geoobjektų, kurių užrašus norite konfigūruoti.

Atidaromas dialogas *Užrašų nustatymai* su šiomis savybėmis. Kaip įprasta, savybės gali būti *priklausomos nuo duomenų*.

10.3.1 Užrašo teksto formatavimas

Whether you are configuring a *Text Format* or *Label Settings* item, and depending on the type of layer you are configuring, you will be given the following options:

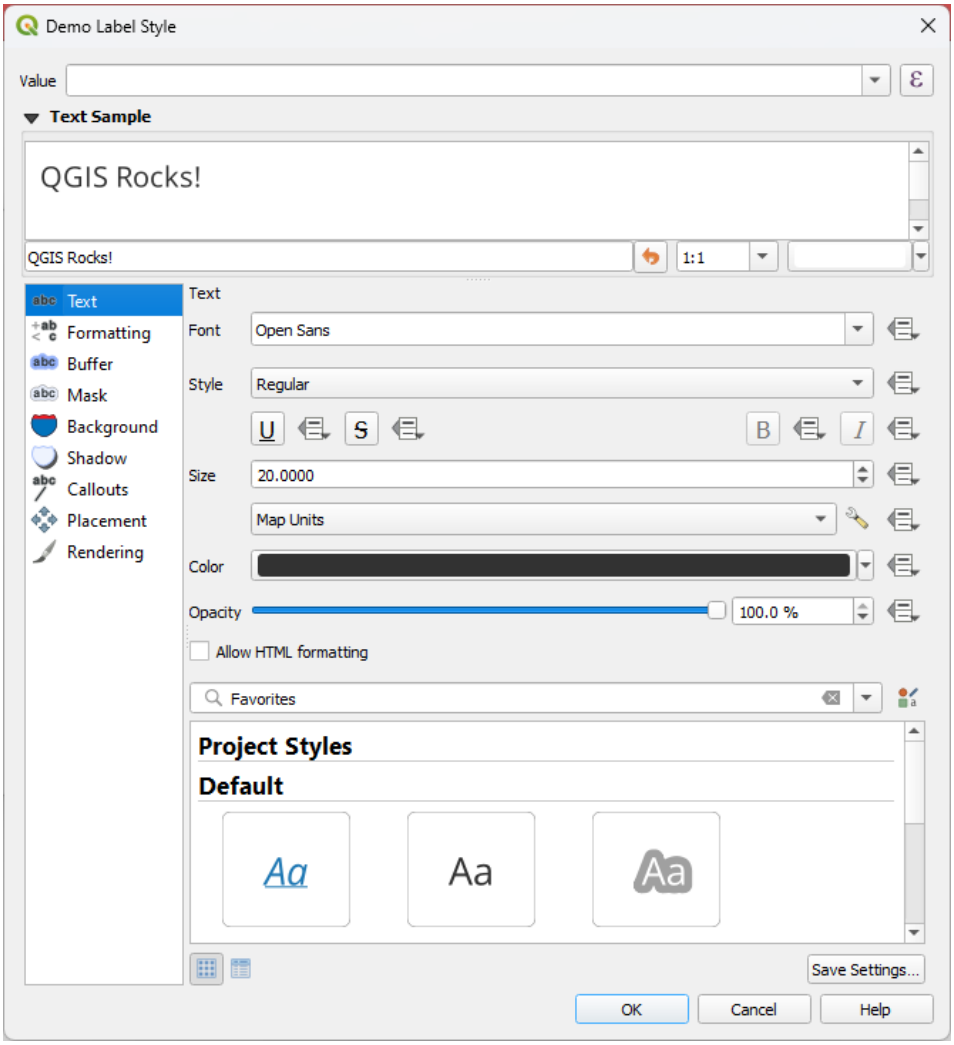
10.4 Table : Label properties by data type

Savybių kortelė	Teksto formatas	Užrašo nustatymai	Vector or mesh layer	raster layer
Tekstas	☑	☑	☑	☑
Formatavimas	☑	☑	☑	☑
Buferis	☑	☑	☑	☑
Slėpimas		☑	☑	
Fonas	☑	☑	☑	☑
Šešėlis	☑	☑	☑	☑
Išnešimas		☑	☑	
Vieta		☑	☑	☑
Braižymas		☑	☑	☑

Dėmesio

While for legibility, „feature“ is the name used below to indicate the item being labeled, depending on the underlying layer type, it can be replaced by „pixel“, „face“ or „vertex“.

Teksto kortelė



10.25 Fig. : Užrašų nustatymai - teksto kortelė

Kortelėje abc *Tekstas* jūs galite nurodyti:

- *Šriftą* iš tų, kurie yra jūsų mašinoje
- *Stilių*: kartu su bendrais šrifto stiliais, jūs galite nurodyti, ar tekstą reikia pabraukti ar perbraukti
- *Dydį* bet kuriais *palaikomais vienetais*
- *Spalvą*
- *Nepermatomumą*
- ir *Leisti HTML formatavimą*, kuris įgalina HTML žymų poaibio naudojimą ir CSS taisykles tinkinant užrašą.

Kortelės apačioje valdiklis rodo filtruojamą sąrašą suderinamų elementų, esančių jūsų *stilių valdymo duomenų bazėje*. Tai leidžia jums lengvai konfigūruoti dabartinį teksto formatą ar užrašo nustatymus pagal jau esamus, taipogi įrašyti naują elementą į stilių duomenų bazę: spauskite mygtuką *Įrašyti formatą...* ar *Įrašyti nustatymus...* ir pateikite pavadinimą ar žymą(as).

Pastaba

Konfigūruojant *Užrašų nustatymų* elementą, teksto formato elementai taipogi pasiekiami šiame valdiklyje. Parinkite vieną, kad greitai perrašytumėte esamo užrašo *tekstines savybes*. Analogišku būdu jūs čia galite sukurti/perrašyti ir teksto formatą.

Leisti HTML formatavimą

Įjungus *Leisti HTML formatavimą*, jums reikia lauke *Reikšmė* pateikti HTML kodą. Naudokite tarpus vietoje tabuliacijos bet kokiam atitraukimui. Išraiška nagrinėjama ir bet kokios palaikomos HTML žymos permuša atitinkamus užrašų savybių nustatymus. Kadangi neįmanoma detaliai išvardinti visų HTML žymų ir CSS savybių, kurias šiuo metu palaiko QGIS, mes kviečiame jus patyrinėti ir išbandyti jūsų užrašuose tas, kurias palaiko naudojama Qt biblioteka.

Palaikomų HTML žymų pavyzdžiai:

- Teksto formatavimas, toks kaip pasvirimas ar riebinimas, pavyzdžiui:

```
<i>QGIS</i> <b>rocks!</b>
```

- Paaukštinimas ar pažeminimas, kur tekstas bus vertikaliai $^{\text{paaukštintas}}$ ar $_{\text{pažemintas}}$ ir automatiškai jo dydis bus 2/3 tėvinio šrifto dydžio. Jūs taipogi galite nustatyti fiksuotą šrifto dydį paaukštinimui/pažeminimui įterpdami css taisykles, pavyzdžiui:

```
<sup style="font-size:33pt">my superscript text</sup>
```

- Horizontalus teksto lygiavimas, naudojant arba HTML atributą `align="xxx"`, arba žymą `center`. HTML5 suderinamumui geriau naudoti CSS savybę `text-align`.

> **Dėmesio**
>
> Horizontalaus lygiavimo negalima naudoti išlenktuose užrašuose.

- Teksto blogo struktūrizavimas naudojant antraštės žymas (tokias kaip `h1`, `h2`, ...) ar pastraipas (su `p`, `div`, `br`):

```
<div class="myDiv">
  <h2>QGIS always rocks!!</h2>
  <p align="center">Let's dive into details of its nice features.</p>
</div>
```

- Piešinių įterpimas: bet kokio formato piešinį, kurį gali perskaityti QGIS, galima įdėti į HTML užrašo turinį. Jis gali būtų teikiamas iš vietinių failų kelių, HTTP nuorodą ar base64 užkoduoto turinio, naudojant atributą `src="xxx"`. Piešinio dydį galima nurodyti atributais `width="##"` ir `height="##"` vienetais `points`. Jei plotis ir aukštis nenurodyti, tai jie bus automatiškai paskaičiuoti pagal pradinį piešinio dydį. Piešiniai gali būti tik įstatyti į vietą, o ne pateikiami kaip plaukiojantys, ir negali būti naudojami išlenktuose užrašuose.

```
<img src="qgis.png" width=40 height=60>
```

Palaikomų CSS savybių pavyzdžiai:

- Šrifto savybės (`color`, `font-family`, `font-size`, `font-weight`, `font-style`, `word-spacing`). Atkreipkite dėmesį, kad `word-spacing` visada naudoja taškų vienetą.
- Teksto dekoracijos, tokios kaip pabraukimas, pabraukimas viršuje ir perbraukimas (`text-decoration`)
- Teksto lygiavimas (`vertical-align`, `text-align`). Horizontalaus lygiavimo negalima naudoti išlenktuose užrašuose.
- Line height, in `points` or `percent` unit, e.g. „line-height: 40pt“ or „line-height: 40%“

- Background properties such as `background-color` and `background-image`. They are supported for block type items (e.g. `div`) or inline items (e.g. `span`). For images, the CSS should be formatted as `background-image: url(xx)` and supports local file paths, HTTP links, or base64 embedded content.

> **Dėmesio**
>
> Backgrounds are not supported for curved text and are always rendered above any background shape for the label, and below drop shadows/buffers properties.

- Margin properties, available for block type items only, such as `div`, `p`, `h1`,... They can be specified in either the longhand or shorthand way, in `points` unit only. Negative values can be set for the bottom margin.

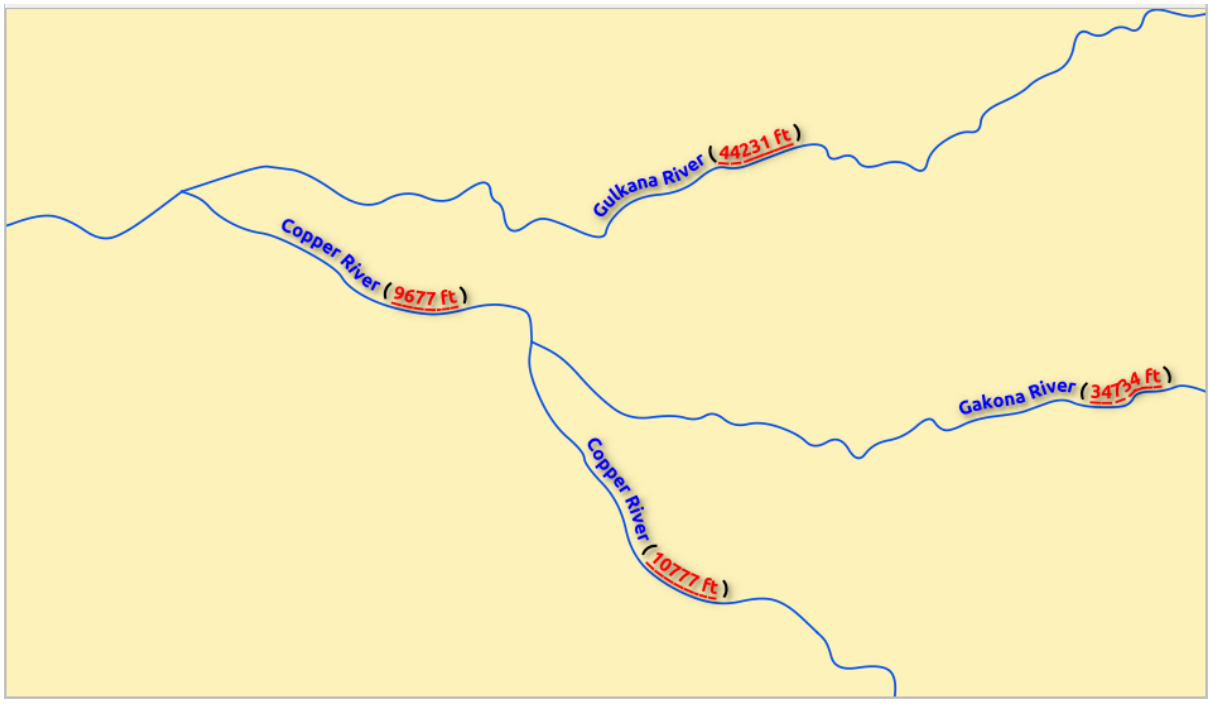
```
<div class="myDiv">
  <h2 style="margin-left: 5pt; margin-right: 10pt">QGIS still rocks...</h2>
  <p style="margin: 5pt 0pt -10pt 0pt">Thanks to you!!</p>
</div>
```

CSS savybes galima nustatyti HTML žymose naudojant atributą `style`. HTML žyma `span` netaikoma jokiam teksto formatavimui ir idealiai tinka, jei tiesiog norite pritaikyti CSS stilių. CSS savybės pavadinimas ir jos reikšmė atskiriama dvitaškiu (`:`). Kelios CSS savybės atskiriamos kabliataškiu (`;`), pvz.:

```
<span style="text-decoration:underline;text-align:center;color:blue;word-spacing:20
↪">I will be displayed as blue underlined and centered text with increased space␣
↪between words</span>
```

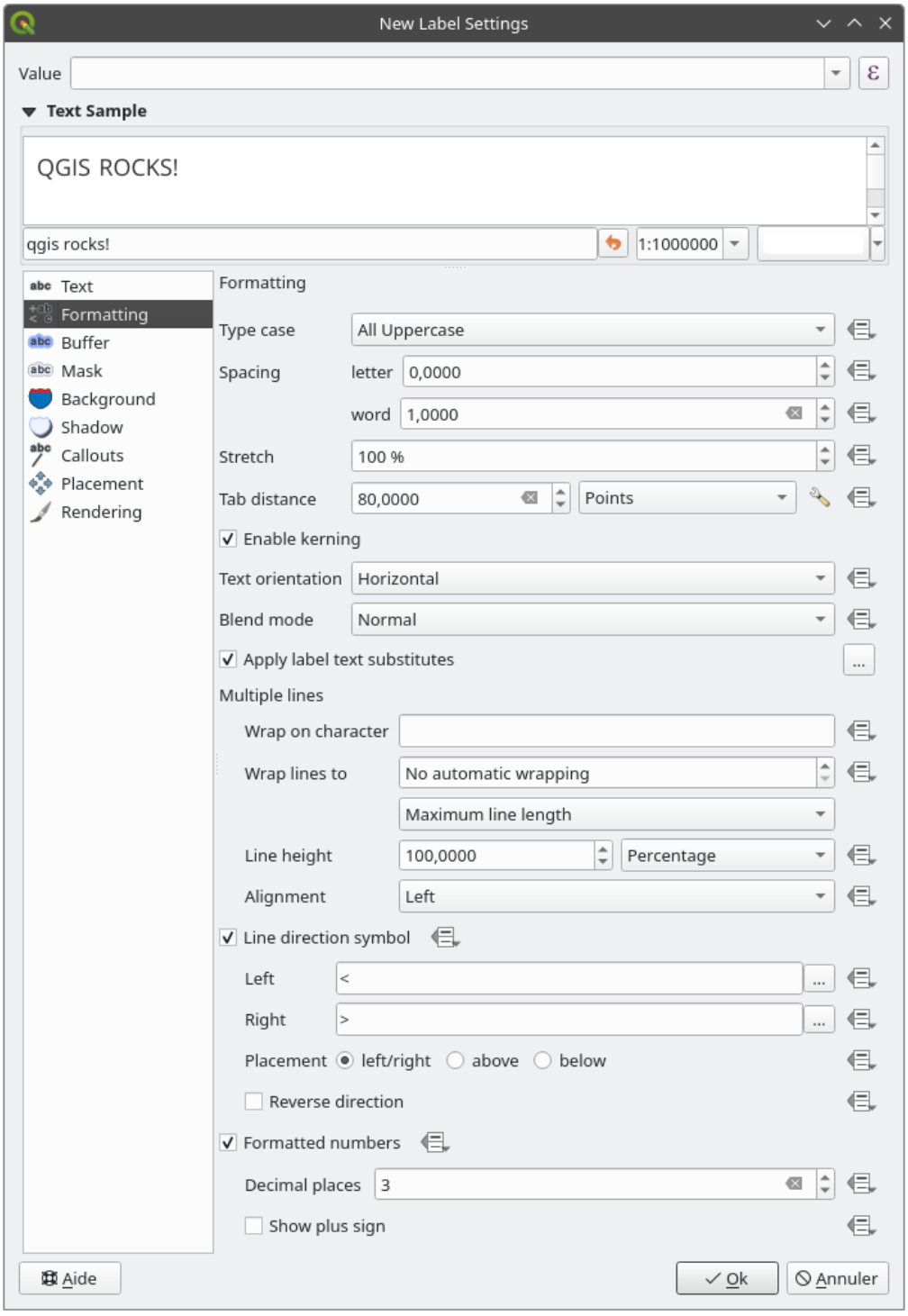
Žemiau pateikiamas pavyzdys HTML paremtos išraiškos ir braižymo (tam pačiam užrašui taikomos skirtingos spalvos ir pabraukimas):

```
format(
  '<span style="color:blue">%1</span> ( <span style="color:red"><u>%2 ft</u></span>
↪ )',
  title( lower( "Name" ) ),
  round($length)
)
```



10.26 Fig. : Užrašai su įjungtu HTML formatavimu

Formatavimo kortelė



10.27 Fig. : Užrašų nustatymai - formatavimo kortelė

Kortelėje *Formatavimas* jūs galite:

- Naudoti parinktį *Raidžių dydis*, kad pakeistumėte teksto raidžių dydį. Turite galimybę rodyti tekstą kaip:
  - *Nekeisti*
  - *Visos didžiosios*

  - *Visos mažosios*
  - *Small Caps*: renders lowercase characters as small caps.
  - *All Small Caps*: renders all characters as small caps regardless of their original case.
  - *Pirmos didžiosios*: pakeičia kiekvieno žodžio pirmą raidę didžiąja, o visas kitas - į mažąsias, jei pradiniame tekste yra vieno dydžio raidės. Jei naudojamos įvairaus dydžio raidės, tai kitos raidės nekeičiamos.
  - *Pirma raidė visada didžioji*: pakeičia pirmą kiekvieno žodžio raidę į didžiąją, o visas kitas palieka tokias, kokios yra.
- *Tarpai* keičia tarpą tarp žodžių ir tarp raidžių.
- *Tabuliacijos atstumas* leidžia keisti `Tab` simbolių pakeitimo atstumą užrašuose bet kuriuo palaikomu vienetu. Tai leidžia, pavyzdžiui, vertikaliai atskirti elementus išdėstymo legendos tekste.

  Pressing the Configure tab stops button on the right, you can add a list of tab positions, instead of a single distance. This allows e.g. the creation of table-like multiline labels, where texts are split over different „columns“ and lined up nicely, based on the `Tab` characters.
- *Ištempimo* koeficientas: leidžia horizontaliai ištempti arba suspausti tekstą nurodytu koeficientu. Patogu tinkinant šriftų pločius, kad įtilptų šiek tiek daugiau teksto.
- *Įjungti kerningą* teksto šriftui
- Nustatykite *Teksto orientaciją*, kuri gali būti *Horizontali* arba *Vertikali*. Ji taipogi gali būti *Paremta pasukimu*, kai nustatomas užrašas (pvz., tinkamai padėti užrašą linijos geoobjektams *lygiagretaus* pozicionavimo režime).
- *Suliejimo režimo* parinktis leidžia nurodyti, kaip jūsų užrašai bus maišomi su po jais esančiais žemėlapio geoobjektais (daugiau informacijos rasite skyriuje *Suliejimo režimai*).
- Parinktis *Naudoti teksto sinonimus* leidžia nurodyti sąrašą tekstų, kuriuos reikia keisti į kitus tekstus geoobjektų užrašuose (pavyzdžiui sutrumpinant gatvių pavadinimus). Pakeitimo tekstai naudojami rodant užrašus žemėlapyje. Naudotojai taipogi gali eksportuoti ir importuoti pakeitimų sąrašus, kad būtų lengviau perpanaudoti ir bendrinti.
- Konfigūruokite *Perkėlimą į kitą eilutę*:
  - Parinktyje *Perkelti ties simboliu* nustatykite simbolį, kuris reikš priverstinį teksto perkėlimą į kitą eilutę.
  - Nustatykite idealų linijos dydį automatiniam perkėlimui į kitą eilutę naudojant parinktį *Kelti eilutes ties*. Dydis gali būti nurodytas arba *Maksimaliu eilutės ilgiu*, arba *Maksimaliu linijos ilgiu*.
  - Nuspręskite, koks turi būti *Eilutės aukštis*: reikšmės gali būti nurodytos *Milimetrais*, *Taškais*, *Pikseliais*, *Procentais* ar *Coliais*. Kai eilutės linijos aukštis nurodomas procentais, tai yra procentai nuo numatytojo tekstinės eilutės tarpo tai šrifto šeimai. Paprastai tai būna nuo 1,2 iki 1,5 teksto dydžio.
  - Nurodykite *Lygiavimą*: tipinės galimos reikšmės yra *Kairėje*, *Dešinėje*, *Lygiuoti* and *Centre*.

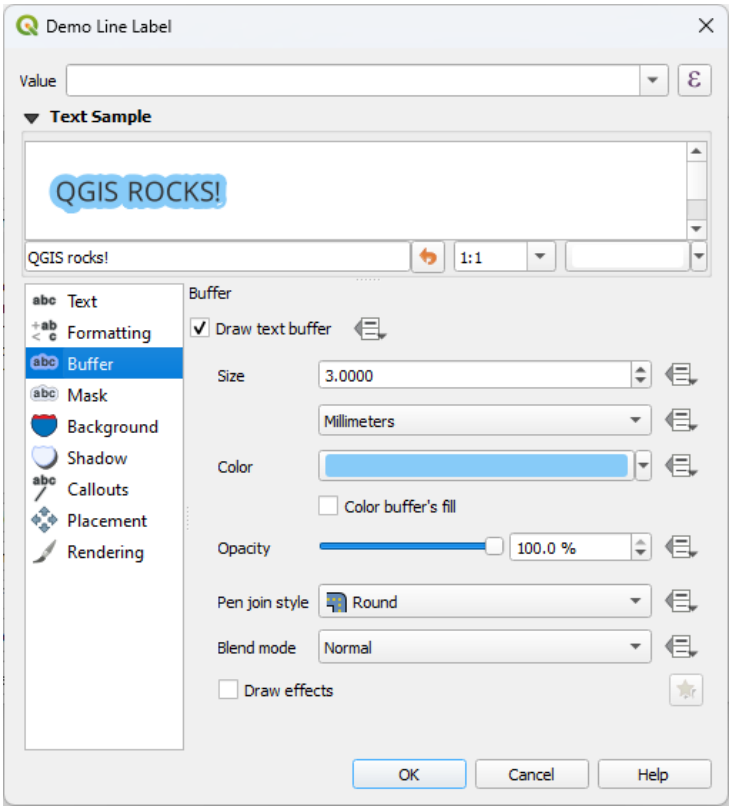
    Nustatant taško užrašų savybes, teksto lygiavimui taipogi galima nurodyti *Sekti užrašo poziciją*. Tokiu atveju lygiavimas priklausys nuo galutinio užrašo vietos santykinai nuo taško. Pavyzdžiui jei užrašas bus taško kairėje, tai užrašas bus lygiuotas pagal dešinį kraštą, o jei užrašas bus dešinėje, tai lygiavimas bus pagal kairįjį kraštą.

> **Pastaba**
>
> *Kelių linijų* formatavimas kol kas nepalaikomas kreivėmis paremtame *užrašų pozicionavime*. Parinktys tada bus neaktyvios.

- Linijų užrašams jūs galite įtraukti *Linijų krypties simbolį*, kad padėtumėte suprasti linijų kryptis simboliais, indikuojančiais *Kairėn* ar *Dešinėn*. Jie veikia ypač gerai, kai naudojami su *išlenktu* ar *lygiagrečiu* pozicionavimu kortelėje *Vieta*. Yra parinktys, leidžiančios nurodyti simbolio vietą bei *Apsukti kryptį*.

- Naudokite parinktį *Formatuoti skaičiai*, kad formatuotumėte skaičių tekstą. Galite nurodyti skaičių *Skaitmenų po kablelio*. Pagal nutylėjimą po kablelio paliekami 3 skaitmenys. Naudokite *Rodyti pliuso ženklą* jei norite, kad teigiamiems skaičiams būtų rodomas pliuso ženklas.

Buferio kortelė



10.28 Fig. : Užrašų nustatymai - buferio kortelė

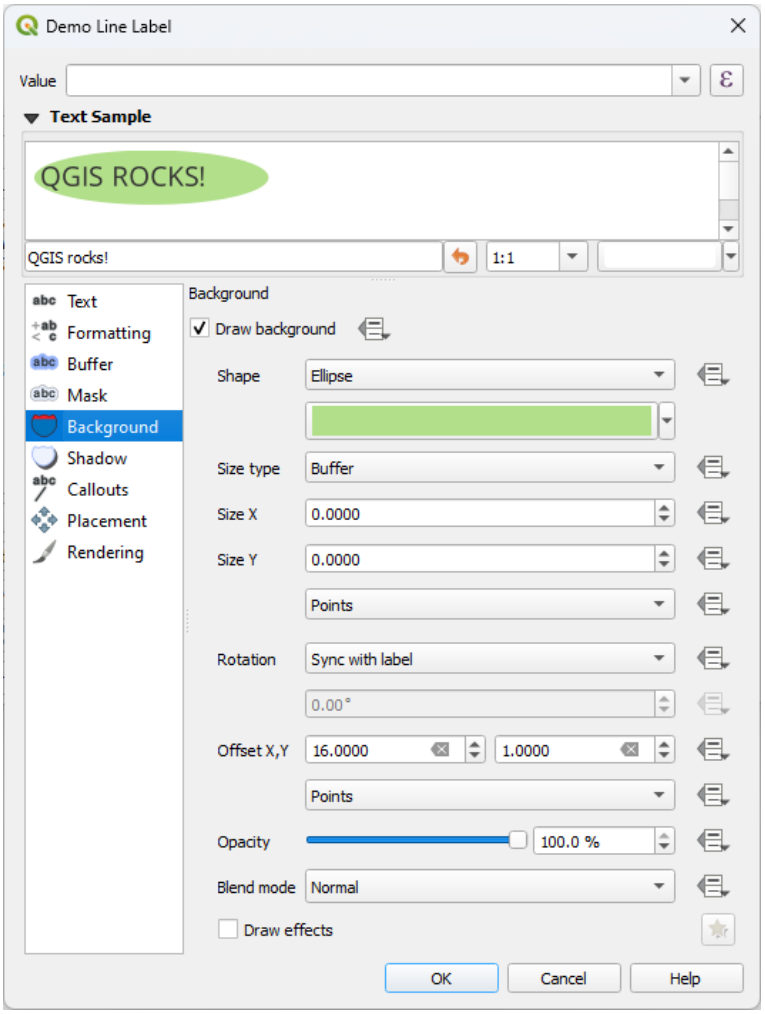
Norėdami sukurti buferį aplink užrašą, įgalinkite parinktį *Paišyti teksto buferį* kortelėje *Buferis*. Tada galėsite:

- Nustatyti buferio *Dydį* bet kuriais *palaikomais vienetais*
- Nustatyti buferio *Spalvą*
- *Buferio pildymo spalva*: Buferis išsiplečia nuo užrašo kontūro, taigi, jei parinktis įjungta, užrašo vidus užpildomas. Tai gali būti svarbu, kai naudojami pusiau permatomi užrašai ar nenormalūs suliejimo režimai, kurie leidžia matyti kiaurai per užrašo tekstą. Išjungus šią parinktį (naudojant visiškai permatomus užrašus) bus galima kurti kontūrais nubraižytus užrašus.
- Nustatyti buferio *Nepermatomumą*
- Taikyti *Piešimo sujungimo stilių*: kuris gali būti *Apvalus*, *Mitra* ar *Kampas*
- Naudoti *Suliejimo* parinktį, kad nurodytumėte, kaip užrašo buferis turi būti maišomas su po juo esančiais žemėlapio komponentais (daugiau informacijos rasite *Suliejimo režimai*).
- Įjungti *Paišymo efektus*, kad pridėtumėte sudėtingesnius *paišymo efektus*, padedančius pagerinti teksto skaitomumą, pavyzdžiui prideda į išorę sklindantį spindėjimą ar išblukinimą.

Fono kortelė

Kortelė *Fonas* leidžia jums konfigūruoti formą, kuri paišoma po kiekvienu užrašu. Kad pridėtumėte foną, įjunkite varnelę *Paišyti foną* ir parinkite *Formą*. Ji gali būti:

- paprasta forma, tokia kaip *Stačiakampis*, *Kvadratas*, *Apskritimas* ar *Elipsė*, naudojant visas *pildymo simbolio* savybes
- *SVG* simbolis iš failo, URL ar įkeltas į projektą ar stilių duomenų bazę (*daugiau informacijos*)
- ar *Žymeklio simbolis*, kurį galite sukurti ar parinkti iš *simbolių bibliotekos*.



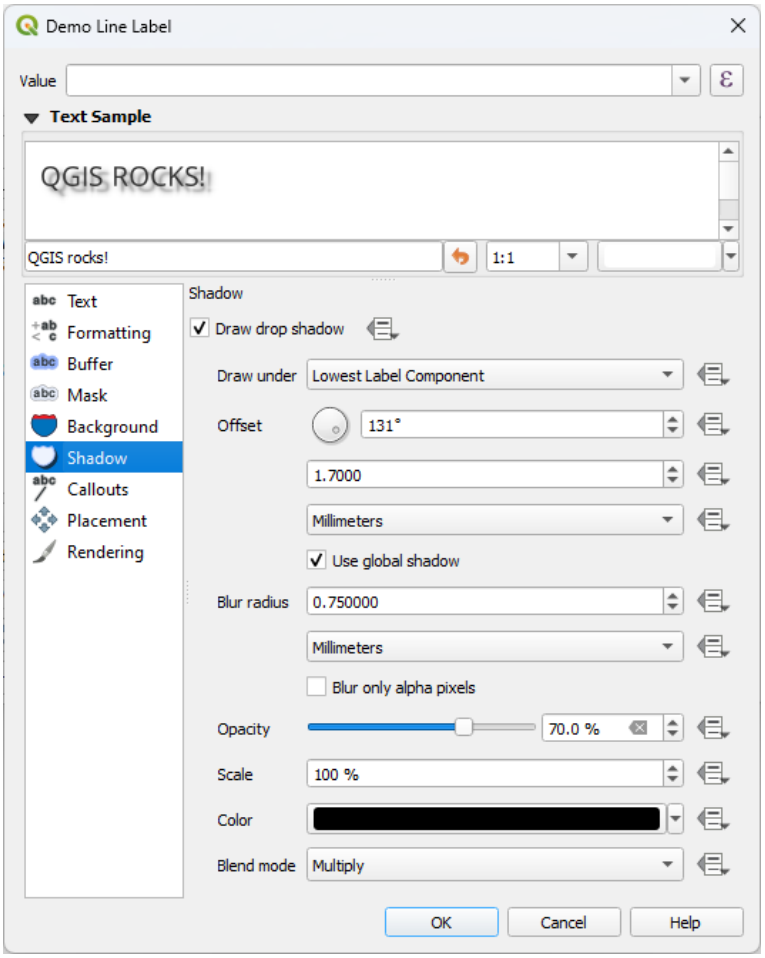
10.29 Fig. : Užrašų nustatymai - fono kortelė

Priklausomai nuo parinktos formos, jums reikia konfigūruoti kai kurias iš šių savybių:

- Rėmo *Dydžio tipą*, kuris gali būti:
  - *Fiksuotas*: naudojant tą patį dydį visiems užrašams, nepriklausomai nuo teksto dydžio
  - arba *Buferis* nuo teksto apimties stačiakampio
- Rėmo *Dydis* X ir Y kryptimis, naudojant bet kokius *palaikomus vienetus*
- Fono *Pasukimas*, gali būti *Sinchronizuoti su užrašu*, *Poslinkis nuo užrašo* ar *Fiksuotas*. Paskutiniams dviems reikia nurodyti kampą laipsniais.
- *Poslinkis X,Y* fono elemento pastūmimui X ir/ar Y kryptimis

- *Spindulys X, Y* - fono formos kampų užapvalinimui (taikoma tik stačiakampio ir kvadrato formoms)
- Fono *Nepermatomumas*
- *Suliejimas* maišo foną su kitais braižomais elementais (žr. *Suliejimo režimai*).
- SVG simboliams galite naudoti jų numatytąsias savybes (*Įkelti simbolio parametrus*) ar nustatyti savo *Pildymo spalvą*, *Brėžimo spalvą* ir *Brėžimo plotį*.
- *Paišyti efektus* leidžia pridėti sudėtingesnius *paišymo efektus* teksto skaitomumo gerinimui, pvz. pridedant išorinį spindėjimą ar suliejimą.

Šešėlių kortelė



10.30 Fig. : Užrašų nustatymai - šešėlių kortelė

Norėdami tekstui pridėti šešėlį, pereikite į kortelę *Šešėliai* ir įjunkite *Paišyti metamą šešėlį*. Tada galėsite:

- Nurodyti elementą, kuris meta šešėlį lauke *Piešti po*. Tai gali būti *Žemiausias užrašo komponentas* arba konkretus elementas, toks kaip pats *Tekstas*, *Buferis* ar *Fonas*.
- Nustatyti šešėlio *Poslinkį* nuo šešėlį metančio elemento, t.y.:
  - Kampą: pagal laikrodžio rodyklę, priklauso nuo po juo esančio elemento orientacijos
  - Poslinkio atstumą nuo šešėlį metančio elemento
  - Poslinkio vienetus

Jei įjungsite varnelę *Naudoti globalų šešėlį*, tai nulinis kampas visada bus šiaurėje ir nepriklausys nuo užrašo elemento orientacijos.

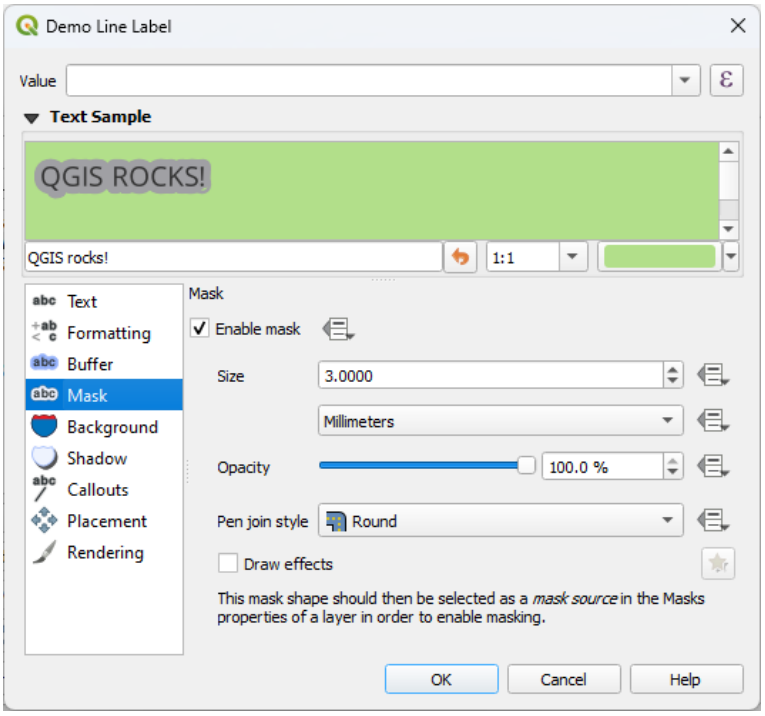
- Įtakoti šešėlio išvaizdą nurodant *Išblukinimo spindulį*. Kuo didesnis šis skaičius, nurodytas jūsų pasirinktais vienetais, tuo švelnesnis bus šešėlis.
- Nurodyti šešėlio *Nepermatomumą*
- Keisti šešėlio dydį naudojant *Mastelio* koeficientą
- Parinkti šešėlio *Spalvą*
- Naudoti *Suliejimo režimo* parinktį, kad nurodytumėte, kaip jūsų užrašo šešėlis turi būti maišomas su po juo esančiais žemėlapio komponentais (daugiau informacijos rasite *Suliejimo režimai*).

10.3.2 Sąveikos su užrašais konfigūravimas

Be teksto formatavimo nustatymų pateiktų čia, jūs taipogi galite nurodyti, kaip užrašai sąveikauja vieni su kitais ar kitais geoobjektais.

Slėpimo kortelė

Slėpimo kortelė leidžia jums nurodyti slėpimo plotą aplink užrašus. Ši savybė labai naudinga, kai turite persidengiančius simbolius ir užrašus su panašiomis spalvomis, o jūs norite, kad užrašai būtų matomi. Užrašo slėpimo plotas neleidžia braižyti nurodytų geoobjektų nurodytu atstumu. Pavyzdžiui jūs galite nurodyti užrašo slėpimą, kad nurodytas sluoksnis nebūtų braižomas 2mm atstumu nuo užrašo, bet kito sluoksnio geoobjektai būtų matomi. Užrašų slėpimas yra panašus į buferius tuo, kad jie leidžia valdyti kitus geoobjektus dengiančių užrašų įskaitomumą. Užrašų buferis braižomas virš po juo esančių geoobjektų, o užrašų slėpimas atrinktinai išima sluoksnius iš braižymo.



10.31 Fig. : Užrašų nustatymai - Slėpimo kortelė (teksto pavyzdys matomas su žaliu fonu, reprezentuojančiu kitą sluoksnį, kuris slepiamas)

Norėdami sukurti užrašų slėpimo efektus:

1. Įjunkite varnelę *Naudoti slėpimą* kortelėje .
2. Tada galėsite nurodyti:
   - slėpimo *Dydį* *palaikomais vienetais*
   - aplink užrašą slepiamo ploto *Nepermatomumą*
   - *Piešimo sujungimo stilių*
   - *paišymo efektus* per varnelę *Paišymo efektai*.
3. Parinkite slėpimo formą kaip slėpimo šaltinį persidengiančių sluoksnių savybių kortelėje *Slėpimas* (žr. *Masks Properties*).

Išnešimų kortelė

Bendra praktika talpinant užrašus tankiame žemėlapyje yra naudoti **išnešimus** - užrašai, kurie yra (ar buvo išnešti) už jų susijusių geoobjektų, identifikuojami dinamine linija, kuri jungia užrašą su geoobjektu. Jei vienas iš dviejų galų (arba prie užrašo, arba prie geoobjekto) perkeliamas, jungties forma perskaičiuojama.

10.32 Fig. : Užrašai su įvairiais išnešimų nustatymais

Norėdami užrašui pridėti išnešimą, įjunkite kortelę *Išnešimai* ir įgalinkite *Paišyti išnešimus*. Tada galėsite:

1. Parinkti jungties *Stilių*, vieną iš:
   - *Paprastos linijos*:tiesi linija, trumpiausias kelias
   - *Manheteno laužytos linijos*: 90° kampais laužyta linija
   - *Lenktos linijos*: kreivės
   - *Balionai*: kalbėjimo balionai apgaubia užrašą ir rodo į geoobjektą. Gali turėti užapvalintus kampus.
2. Nurodyti išnešimo savybes. Šioje lentelėje išvardintos skirtingos savybės su aprašymu ir suderinamu jungties stiliumi.

10.5 Table : Užrašo išnešimo savybės

Savybė	Išnešimo stilius	Aprašymas
Pildymo stilius	Balionai	*Pildymo simbolis* su pilnomis rodymo galimybėmis, įskaitant sluoksnio efektus, nuo duomenų priklausančius nustatymus, … baliono formos paišymui.
Kampo spindulys		Kalbos burbulio kampo spindulys
Pleišto plotis		Nustato, kokio dydžio turi būti kalbos burbulo jungtis su geoobjekto rodykle
Paraštės		Paraštės aplink užrašo tekstą, jūsų pasirinktais vienetais
Linijos stilius	Visi išskyrus balionus	*Linijos simbolis* su pilnomis rodymo galimybėmis, įskaitant sluoksnių efektus, nuo duomenų priklausančius nustatymus, … jungties linijos paišymui.
Išlenktumas	Lenktos linijos	Jungties linijos išlenktumo procentai
Orientacija		Orientacija, pradedant nuo užrašo iki geoobjekto. Gali būti *Pagal laikrodžio rodyklę*, *Prieš laikrodžio rodyklę* ar *Automatinis* (kiekvienam užrašui nustatoma optimali orientacija).
Minimalus ilgis	Visi išskyrus balionus	Minimalus jungties linijos ilgis
Braižyti visų geoobjekto dalių linijas		Nurodo, ar kelių dalių geometrijos atveju jungties linijas turi būti braižoma nuo užrašo iki kiekvienos geometrijos dalies.
Užrašų pririšimo taškas		Nurodo, kur jungties linijas turi jungtis prie užrašo teksto. Galimos parinktys: • *Artimiausias taškas* • *Centroidas* • Fiksuota kraštinė pozicija (*Viršuje kairėje*, *Viršuje centre*, *Viršuje dešinėje*, *Kairėje per vidurį*, *Dešinėje per vidurį*, *Apačioje kairėje*, *Apačioje centre* ir *Apačioje dešinėje*).
Atitraukti nuo užrašo zonos		Valdo atstumą nuo užrašo pririšimo taško (kur baigiasi išnešimo linija). Tai leidžia išvengti linijos brėžimo iki pat teksto.
Atitraukti nuo geoobjekto	Visi	Valdo atstumą nuo geoobjekto (ar jo pririšimo taško, jei tai poligonas), kur baigiasi išnešimo linija. Pavyzdžiui tai išvengia linijų braižymo iki pat geoobjekto krašto.
Geoobjekto pririšimo taškas		Kur ant (poligono) geoobjekto baigiasi jungties linija. Galimos parinktys: • *Nepasiekiamumo taškas* • *Taškas išorėje* • *Taškas ant paviršiaus* • *Centroidas*
Pabaigos taško žymeklis	Balionai	*Žymeklio simbolis* su pilnomis rodymo galimybėmis, įskaitant sluoksnių efektus, nuo duomenų priklausančius nustatymus, … žymeklio simbolio braižymui po baliono išnešimo pabaigos tašku.
Suliejimas	Visi	Valdo išnešimo *suliejimą*.

Nuo duomenų priklausančios vietos grupėje, galima valdyti koordinates jungties *Pradžios* (užrašo pusėje) ir/ar *Pabaigos* (geoobjekto pusėje) taškų. Jungtis galima valdyti ir rankiniu būdu naudojant įrankį Perkelti užrašą, diagramą ar išnešimą, kurį rasite *Užrašų įrankinėje*. Tokiu būdu galima perkelti kiekvieno išnešimo pradžios ir pabaigos taškus. Viršūnes reikia pažymėti, kai greta yra pelės žymeklis. Jei reikia, perkeliant galima laikyti mygtuką `Shift`. Taip taškas bus pritrauktas taip, kad kampas tarp dviejų išnešimo taškų dalinsis iš 15 laipsnių.

Vietos kortelė

Pasirinkite kortelę *Vieta*, jei norite konfigūruoti užrašo vietą ir užrašų prioritetą. Pastebėtina, kad vietos parinktys skiriasi priklausomai nuo vektorinio ar tinklelio sluoksnio tipo, įvardintus taškus, linijas ar poligonus įtakoja globalūs *PAL nustatymai*.

Taškų sluoksnių vietos nustatymas

Taškų užrašų vietų nustatymo režimai yra:

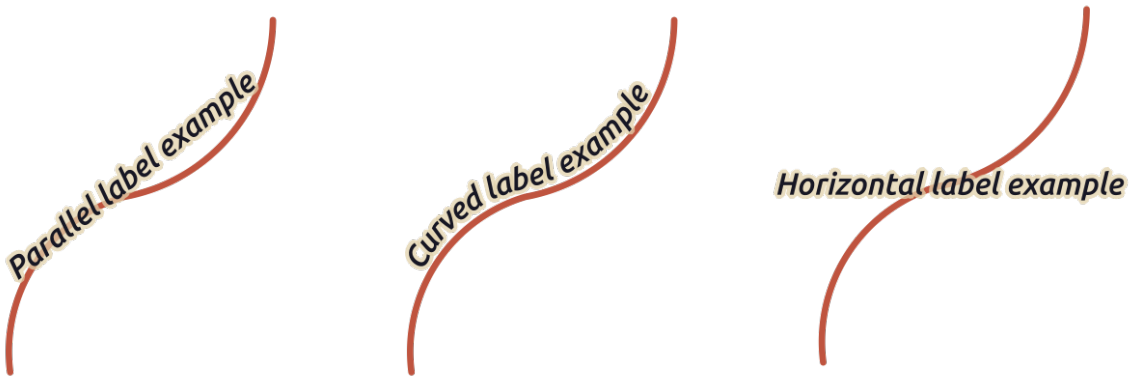
- *Kartografinis*: taškų užrašai generuojami su geresniu vizualiu ryšiu su taškiniu geoobjektu pagal idealios kartografinės vietos taisykles. Užrašus galima padėti:
  - nurodytu *Atstumu palaikomais vienetais*, arba nuo taškinio geoobjekto, arba nuo geoobjektui atvaizduoti parinkto simbolio (nurodoma lauke *Poslinkis matuojamas nuo*). Paskutinė parinktis ypatingai naudinga, kai simbolio dydis nėra fiksuotas, t.y. jis nustatomas priklausomai nuo duomenų ar kai naudojami skirtingi *kategorizuoto* braižymo simboliai.
  - per *Maksimalų atstumą* nuo geoobjekto, tai neprivalomas nustatymas, leidžiantis valdyti kiek toli užrašas gali būti nuo geoobjekto. Tai veikia kartu su užrašo vietos paieškos nustatymu *Atstumas*, pridedant galimybę efektyviau rasti užrašų pozicijas, ypač stipriai užpildytuose žemėlapiuose, užtikrinant, kad jie tiksliai atsiduria greta atitinkamų geoobjektų.
  - naudoti parinktį *Prioritetizuoti padėti*, kuri nusprendžia, kas svarbiau ieškant užrašo vietos. Yra dvi parinktys:
    - *Pirmenybė artimesniems užrašams*: Pagal nutylėjimą užrašus stengiamasi laikyti kiek galima arčiau geoobjekto.
    - :guilabel:`Pirmenybė vietos rikiavimui: Užrašas stengsis likti konkrečioje vietoje (pavyzdžiui viršuje kairėje arba viršuje dešinėje), net jei jis šiek tiek toliau nuo geoobjekto. Užrašas bus perkeltas tik jei jūsų norimoje pozicijoje nėra vietos maksimaliu atstumu.
  - pagal *Pozicijos prioritetus*, kurie diktuoja vietos kandidatus užrašų pririšimui aplink ir (centruotai) virš taškų geoobjektų bei šių pozicijų tikrinimo rikiuotę. Numatytoji rikiuotė, kuri remiasi Krygier ir Wood instrukcijomis (2011) ir kitais kartografijos vadovėliais, yra tokia:
    1. viršuje dešinėje
    2. viršuje kairėje
    3. apačioje dešinėje
    4. apačioje kairėje
    5. per vidurį dešinėje
    6. per vidurį kairėje
    7. viršuje, šiek tiek dešiniau
    8. apačioje, šiek tiek kairiau.

    Naudodami mygtuką Nuo duomenų priklausantis permušimas jūs galite pateikti kableliais atskirtą sąrašą vietų surikiuotų pagal prioritetą. Tai leidžia kai kuriems objektams naudoti tik kai kurias vietas, tai pavyzdžiui kranto geoobjektų objektai gali išvengti užrašų virš žemės.
- *Aplink tašką*: užrašai išdėstomi apskritimu aplink geoobjektą lygiu spinduliu, nurodytu *Atstumu*. Papildomai galite nurodyti *Maksimalų atstumą* nuo geoobjekto, kad valdytumėte kaip toli užrašas gali būti nuo geoobjekto. Pozicijos prioritetas yra pagal laikrodžio rodyklę pradedant nuo „viršuje dešinėje“. Poziciją galima apriboti naudojant nuo duomenų priklausančią parinktį *Kvadrantas*.
- *Poslinkis nuo taško*: užrašai išdėstomi įvairiais vienetais nurodytu *Poslinkio X,Y* atstumu nuo taško geoobjekto arba virš geoobjekto. Jūs galite naudoti nuo duomenų priklausantį *Kvadrantą*, kad apribotumėte vietą ar priskirtumėte užrašui *Pasukimą*.

Linijų sluoksnių vietos nustatymas

Linijų sluoksnių užrašų režimai yra:

- *Lygiagrečiai*: braižo užrašą lygiagrečiai generalizuotos linijos išraiškos, stengiantis užrašą padėti virš lygių linijos dalių. Jūs galite nurodyti:
  - *Leidžiamas pozicijas*: *Virš linijos*, *Anti linijos*, *Po linija* ir *Vieta priklauso nuo linijos padėties* (užrašą išdėstant linijos kairėje ar dešinėje pusėje). Galima vienu metu parinkti kelias parinktis. Tokiu atveju QGIS ieškos optimalios užrašo vietos.
  - *Atstumą* tarp užrašo ir linijos
- *Išlenktai*: braižo užrašą sekant linijos geoobjekto iškraipymus. Be parametrų, kurie naudojami ir *Lygiagrečiame* režime, jūs galite nurodyti *Maksimalų kampą tarp išlenktų simbolių*, viduje arba išorėje.
- *Horizontaliai*: braižo užrašus horizontaliai išilgai linijinio geoobjekto.



10.33 Fig. : Linijų užrašų išdėstymo pavyzdžiai

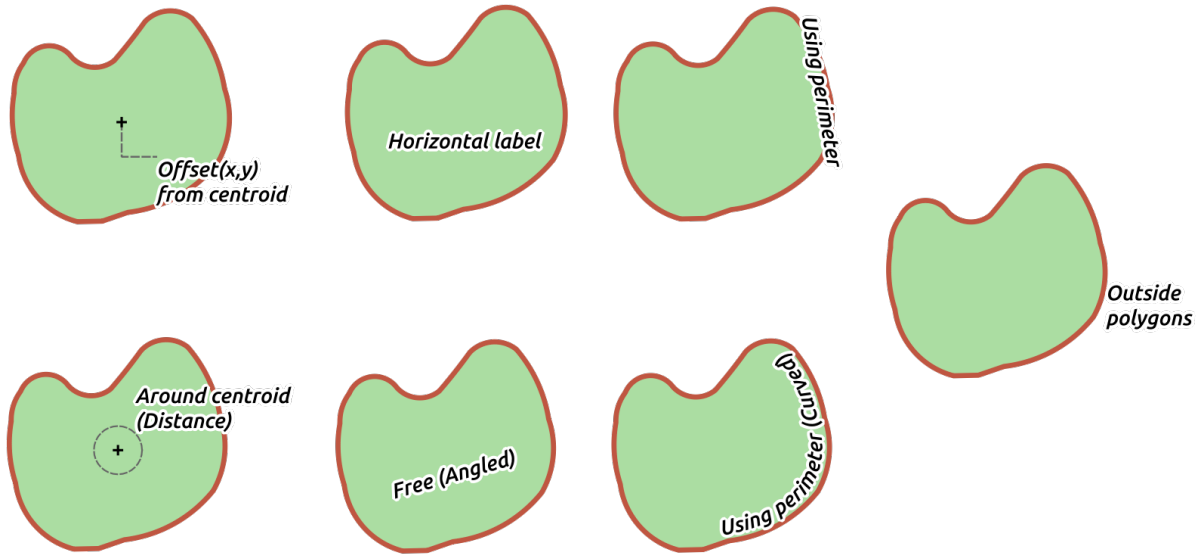
Be vietos nustatymo režimo, jūs galite nurodyti:

- *Užrašų pasikartojimai Atstumas* po kurio galima pakartotinai rodyti užrašą išilgai geoobjekto ilgio. Atstumą galima nurodyti `Milimetrais`, `Taškais`, `Pikseliais`, `Metrais mastelyje`, `Žemėlapio vienetais` ir `Coliais`.
- *Užrašų išsikišimas Atstumas* (netaikomas horizontaliame režime): nurodo maksimalų leidžiamą atstumą, kuriuo užrašas gali išsikišti (ar prasidėti) už linijos geoobjekto. Didinant šią reikšmę galima leisti rodyti užrašus trumpesniems linijiniams geoobjektams.
- *Užrašų susiejimas*: valdo užrašų vietą išilgai susijusio geoobeketo linijos. Spauskite *Nustatymai* ..., kad pasirinktumėte:
  - poziciją išilgai linijos (kaip koeficientą), arti kurios bus išdėstomi užrašai. Tai gali būti ir nuo duomenų priklausantis nustatymas su galimomis reikšmėmis:
    - *Linijos centre*
    - *Linijos pradžioje*
    - *Linijos pabaigoje*
    - ar *Savas...*.
  - *Apkarpymas*: nurodo kaip skaičiuojama vieta ant linijos. Pagal nutylėjimą naudojama tik matoma linijos dalis, bet stabilesniems rezultatams galima naudoti pilną apimtį.
  - *Teksto pririšimo taškas*: nurodo kuri teksto dalis (pradžia, vidurys ar pabaiga) bus sulygiuota su pririšimo tašku. Naudojant *Automatinį* pririšimą:

    * Užrašams, kurie pririšti prie linijos pradžios (0-25%), pririšimo vieta bus užrašo teksto **pradžia**
    * Užrašams, kurių pririšimas yra prie linijos pabaigos (75-100%), pririšimo vieta bus užrašo teksto **pabaiga**
    * Užrašams, kurie pririšti prie linijos centro (25-75%), pririšimo vieta bus užrašo teksto **vidurys**
  - *Vietos elgsena*: naudokite *Pageidaujamą vietos užuominą*, kad užrašo pririšimas būtų traktuojamas tik kaip užrašo vietos nustatymo užuomina. Jei pasirinksite *Griežta*, užrašai bus braižomi būtent užrašo pririšimo vietoje.

Poligonų sluoksnių vietos nustatymas

Galite pasirinkti vieną iš šių poligonų užrašų vietos nustatymo režimų:



10.34 Fig. : Poligonų užrašų vietos nustatymų pavyzdžiai

- *Poslinkis nuo centroido*: užrašai išdėstomi virš geoobjekto centroido arba nurodytu *Poslinkio X,Y* atstumu (*palaikomais vienetais*) nuo centroido. Atskaitos centroidas nustatomas pagal žemėlapio drobėje braižomo poligono dalį (*matomas poligonas*) arba *visą poligoną*, nepriklausomai nuo to, ar jūs galite jį matyti. Jūs taipogi galite:
  - priversti, kad centroidas būtų poligono viduje
  - užrašą išdėstyti konkrečiame kvadrante
  - priskirti pasukimą
  - *Leisti išdėstyti užrašus už poligono ribų*, kai neįmanoma juos išdėstyti poligono viduje. Nuo duomenų priklausomų savybių dėka galima leisti arba neleisti užrašus už poligono ribų ar net priversti užrašus už geoobjekto ribų skirtingai kiekvienam konkrečiam geoobjektui.
- *Aplink centroidą*: išdėsto užrašą nurodytu atstumu aplink centroidą teikiant prioritetą išdėstymui tiesiai virš centroido. Vėlgi, jūs galite nurodyti, ar centroidas yra *matomo poligono* ar *viso poligono*, taipogi ar reikia priversti, kad centroido taškas būtų poligono viduje.
- *Horizontaliai*: užrašas padedamas geriausioje horizontalioje pozicijoje poligono viduje. Stengiamasi užrašą padėti toliau nuo poligono kraštų. Galima *Leisti užrašus padėti už poligono ribų*.
- *Laisvas (pakreiptas)*: užrašas padedamas geriausioje vietoje pasuktas poligono viduje. Pasukimas atsižvelgia į poligono orientaciją ir stengiamasi parinkti poziciją toliau nuo poligono kraštų. Galima *Leisti užrašus padėti už poligono ribų*.

- *Naudojant perimetrą*: braižo užrašą išilgai generalizuotos poligono kontūro linijos, prioritetą teikiant tiesesnėms perimetro dalims. Jūs galite nurodyti:
  - *Leidžiamas pozicijas*: *Virš linijos*, *Anti linijos*, *Po linija* ir *Vieta priklauso nuo linijos padėties* (užrašą išdėstant poligono kontūro kairėje ar dešinėje pusėje). Galima vienu metu parinkti kelias parinktis. Tokiu atveju QGIS ieškos optimalios užrašo vietos.
  - *Atstumą* tarp užrašo ir poligono kontūro
  - *Pasikartojančių užrašų Atstumą*, kad užrašas būtų palei perimetrą braižomas kelis kartus.
- *Naudojant perimetrą (išlenktai)*: braižo užrašą pagal poligono ribos iškraipymą. Be parametrų, kuriuos galima rasti režime *Naudoti perimetrą*, jūs galite nurodyti *Maksimalų kampą tarp išlankstytų simbolių*, viduje arba išorėje.
- *Už poligonų*: užrašus visada išdėsto už poligonų nurodytu *Atstumu*

Bendri vietos nustatymai

Kai kurie užrašų vietos nustatymai taikomi visų sluoksnių geometrijų tipams:

Label Spacing

The *Label Spacing* option allows you to set the margin around the label, preventing all other labels from being placed within that area. You can dynamically control *Margin around label* using the Data-defined override button. Choose unit from the unit selector, and there you can also *adjust scaling range*.

Dėmesio

This option is only available on QGIS installed with GEOS >= 3.10 (see *Help* ► *About* menu).

Duplicate Labels

Use *Avoid duplicate labels* to improve label appearance in situations like road networks, where features such as dual carriageways or service roads can result in many identical labels appearing close together. When active, it prevents placing labels with the exact same text (case-sensitive) closer than the minimum distance defined in the *Settings…* option. The setting applies across layers, so duplicate text from any layer will be considered.

Dėmesio

This option is only available on QGIS installed with GEOS >= 3.10 (see *Help* ► *About* menu).

Geometrijos generatorius

Geometrijos generatoriaus skiltis leidžia naudotojui keisti geometriją, kuri naudojama užrašo pozicionavimui ir braižymui, naudojant *išraiškas*. Tai gali būti naudinga, norint dinamiškai paslinkti geometriją ar pakeisti ją į kitą geometriją (tipą).

Norėdami naudoti geometrijos generatorių:

1. Įjunkite parinktį *Geometrijos generatorius*
2. Įveskite išraišką, remiantis kuria bus generuojama geometrija

3. Jei reikia, parinkite išraiškos išvesties geometrijos tipą: nuo geometrijos priklausantys užrašų nustatymai, tokie kaip vieta ar braižymas, atnaujinami, kad atitiktų naujo geometrijos tipo galimybes.

Keli iš galimų panaudojimo atvejų:

- Naudoti kitą geometriją, kuri įrašyta kitame lauke „užrašo_pozicija“
- Naudoti simbologijoje *sugeneruotą geometriją* užrašų išdėstymui
- Naudoti kintamąjį @map_scale skaičiuojant atstumus / dydžius, kurie nepriklausytų nuo mastelio.
- Kartu su iškraipytos pozicijos režimu sukuria apskritimu aplink taško geoobjektą išdėstytą užrašą:

```
exterior_ring(make_circle($geometry, 20))
```

- Pridėti užrašą linijos geoobjekto pradžioje ir pabaigoje:

```
collect_geometries( start_point($geometry), end_point($geometry) )
```

- Remtis glotnesne upės linija, kad būtų gauta daugiau vietos užrašui:

```
smooth( $geometry, iterations:=30, offset:=0.25, min_length:=10 )
```

Nuo duomenų priklausantis

Grupė *Nuo duomenų priklausantis* teikia tiesioginį užrašų vietos nustatymo valdymą kiekvienam geoobjektui atskirai. Remiantis atributais ar išraiška nustatoma:

- *X* ir *Y* koordinatė
- teksto lygiavimas virš savo pozicijos, nustatytos aukščiau:
  - *Horizontaliai*: gali būti **Kairėje**, **Centre** ar **Dešinėje**
  - *Vertikaliai*: gali būti **Apačioje**, **Pagrindas**, **Pusė**, **Kepurė** ar **Viršuje**
- teksto *Pasukimas*. Pasukimas apibrėžiamas kaip pagal laikrodžio rodyklę skaičiuojamas kampas, kurio 0° rodo į rytus `Horizontaliai` orientuotam tekstui ir į šiaurę `Vertikaliai` orientuotam tekstui. Užrašų pasukimui galima apibrėžti skirtingus vienetus (pvz. `laipsniai`, `lanko minutės`, `apsisukimai`). QGIS pagal nutylėjimą teikia prioritetą ekrano rodymo pasukimo kampams ir visada pasuka tekstą taip, kad jis būtų geriausiai įskaitomas ekrane, taigi „aukštyn-kojų“ kampai bus permušti braižymo metu. Jei norite pakeisti šią elgseną ir priversti QGIS braižyti būtent pagal duomenis paskaičiuotą kampą, yra parinktis *Braižymo* kortelėje, *Rodyti aukštyn kojomis apverstus užrašus*. Įjunkite *Išlaikyti pagal duomenis paskaičiuotas pasukimo reikšmes*, jei norite išlaikyti pasukimo reikšmę susijusiame lauke ir pritaikyti ją užrašui, nepriklausomai nuo to užrašas prisegtas ar ne. Išjungus šią parinktį atsegus atstatomas užrašo pasukimas ir reikšmė išvaloma iš atributų lentelės.

> **Pastaba**
>
> Nuo duomenų priklausantis poligono geoobjektų pasukimas šiuo metu palaikomas tik su vietos režimu *Aplink centroidą*.

> **Pastaba**
>
> Išraiškų negalima naudoti kartu su užrašų žemėlapių įrankiais (t.y. įrankiais *Pasukti užrašą* ir *Perkelti užrašą*) užrašų vietą keičiant *priklausomai nuo duomenų*. Valdiklis bus grąžintas į atitinkamą *papildomą saugyklos lauką*.

Prioritetas

Skiltyje *Prioritetas* jūs galite kiekvienam užrašui nurodyti vietos prioriteto lygį, t.y. jei yra toje pačioje vietoje yra kelios diagramos ar užrašų kandidatai, bus rodomas elementas su aukštesniu prioritetu, o kiti bus praleisti.

Prioriteto lygis taipogi naudojamas įvertinant, ar užrašą galima praleisti dėl aukštesnio įverčio *kliūties geoobjekto*.

Kliūtys

Kai kuriose situacijose (pvz. esant dideliam užrašų tankumui, persidengiant geoobjektams…) užrašams gali būti parinktos vietos virš nesusijusių geoobjektų.

Kliūtis yra geoobjektas, virš kurio QGIS vengia išdėstyti geoobjektų užrašų ar diagramų. Tai galima valdyti skiltyje *Kliūtys*:

1. Įjunkite parinktį *Geoobjektai kaip kliūtys*, kad nuspręstumėte, kad sluoksnio geoobjektai turi veikti kaip kliūtys bet kokiam užrašui ar diagramai (įskaitant ir kitiems to paties sluoksnio geoobjektams).

   Vietoje viso sluoksnio jūs galite parinkti tik poaibį geoobjektų, kuriuos reikia naudoti kaip kliūtis, naudojant greta parinkties esantį valdiklį Nuo duomenų priklausantis permušimas.

2. Naudokite mygtuką *Nustatymai*, kad tinkintumėte kliūčių svorius.

   - Kiekvienam potencialiam kliūties geoobjektui jūs galite priskirti *Kliūties svorį*: bet koks *užrašas* ar *diagrama*, kurios vietos prioriteto lygis yra didesnis už šią reikšmę gali būti perrašytas. Užrašai ir diagramos su žemesniu lygmeniu bus praleisti, jei jokia kita pozicija nėra įmanoma.

     Šie svoriai taipogi gali priklausyti nuo duomenų, kad būtų labiau tikėtina, jog viename sluoksnyje kai kurie geoobjektai dengtų kitus.

   - Poligonų sluoksniams jūs galite parinkti, kokio tipo kliūtis yra geoobjektas:

     – **virš geoobjekto vidaus**: vengia padėti užrašus virš vidinės poligono dalies (stengiasi užrašus padėti pilnai už arba tik truputį poligono viduje)

     – arba **virš geoobjekto ribos**: vengia išdėstyti užrašus virš poligono ribos (stengiantis padėti užrašus už poligono ribų arba pilnai jo viduje). Tai gali būti naudinga sluoksniams, kur geoobjektai dengia visą plotą (administracinius vienetus, kategorines konvergencijas, …). Tokiu atveju neįmanoma išvengti išdėstyti užrašus šių geoobjektų viduje ir vaizdas būna daug geresnis, jei vengiama juos išdėstyti ant geoobjektų ribų.

Braižymo kortelė

Kortelėje *Braižymas* jūs galite nurodyti, kada reikia braižyti užrašus ir jų sąveiką su kitais užrašais ir geoobjektais.

Užrašų parinktys

Skiltyje *Užrašų parinktys*:

- Jūs rasite *nuo mastelio* ir *Pikselio dydžio* priklausančius matomumo nustatymus.
- *Užrašo z-indeksas* nurodo užrašų braižymo rikiuotę bei kurie užrašai braižomi atsižvelgiant į kitus to paties sluoksnio geoobjektų užrašus (naudojant nuo duomenų priklausančias permušimo išraiškas), bei kitų sluoksnių užrašus. Užrašai su didesniu z-indeksu braižomi virš kitų (bet kokio sluoksnio) užrašų su mažesniu z-indeksu.

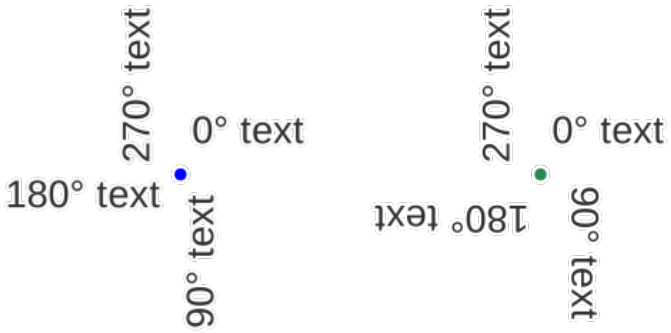
  Be to, logika buvo patinkinta taip, kad jei du užrašai turi tokį patį z-indeksą, tada:

  – jei jie yra tame pačiame sluoksnyje, mažesnis užrašas bus braižomas virš didesnio užrašo

  – jei jie yra iš skirtingų sluoksnių, užrašai bus braižomi tokia tvarka, kaip jų sluoksniai (t.y. atsižvelgiant į jų rikiuotą žemėlapio legendoje).

> **Pastaba**
>
> Šis nustatymas nereiškia, kad užrašai braižomi po kitų sluoksnių geoobjektais, jis tik valdo rikiuotę, kaip užrašai braižomi virš visų sluoksnių geoobjektų.

- *Leisti prastas atsargines pozicijas*: Pagal nutylėjimą QGIS stengiasi braižyti užrašus jų geriausiose pozicijose pagal jūsų nustatymus. Įjunkite šį režimą, kad leistumėte užrašams atsidurti prastesnėse pozicijose, kai nėra kito pasirinkimo, pvz. kai linija yra per trumpa, kad joje tilptų iškreiptas užrašas, kuris turi būti braižomas horizontaliai tik aplink geoobjekto centrinį tašką.
- Naudojant nuo duomenų priklausančius nustatymus *Rodyti užrašą* ir *Visada rodyti*, jūs galite tinkinti, kurie užrašai rodomi.
- Leisti *Rodyti užrašus aukštyn-kojomis*: alternatyvos yra **niekada**, **kai nurodytas pasukimas** arba **visada**.
  - **niekada** - numatytoji reikšmė, prioritetizuojamas ekrano skaitomumas,
  - **kai nurodytas pasukimas** - užrašų pasukimas turi būti nurodytas kortelėje *Vieta*, grupėje *Nuo duomenų priklausantis*
  - **visada** - leidžiami užrašai aukštyn-kojomis



10.35 Fig. : Nuo duomenų priklausantis pasukimas su užrašų rodymo aukštyn-kojomis parinktimi nustatyta: „niekada“ (kairėje), „kai nurodytas pasukimas“ (dešinėje)

- Grupė *Persidengiantys užrašai* leidžia jums valdyti, ar sluoksnyje leidžiami persidengiantys geoobjektų užrašai ir ką reikia daryti su kiekvienu iš jų:
  - *Niekada nepersidengia*: niekada neleisti sluoksnio užrašams persidengti, net jei tai reiškia, kad kai kurių užrašų trūks
  - *Leisti persidengimus, jei reikia*: jei nėra kitos galimybės padėti užrašą, braižyti persidengiantį užrašą. Šiame režime užrašai, jei tik bus galimybė, bus perkelti į ne tokią idealią poziciją, pvz. užrašas bus perkeltas toliau nuo linijos ar poligono centro, JEI tai padės išvengti užrašų persidengimo. Bet jei nėra kitos išeities, tada bus braižomas persidengiantis užrašas.
  - *Leisti persidengimą be baudų*: Nesvarbu, ar užrašas dengia kitą užrašą ar kliūtį, geriausia vieta (pvz. labiausiai centrinė pozicija) turi būti visada naudojama, net jei yra tolimesnė alternatyvi pozicija, kuri išvengtų persidengimo.

  Leidus tiek persidengiančius užrašus, tiek atsargines parinktis, bus garantuota, kad visų geoobjektų užrašai bus nubraižyti… nebūtinai sukuriant geriausią vaizdą!

Geoobjektų parinktys

Skiltyje *Geoobjektų parinktys*:

- Jūs galite pasirinkti *Žymėti visas kelių dalių geoobjektų dalis* ir *Riboti pažymėtų geoobjektų skaičių*.
- Tiek linijų, tiek poligonų sluoksniai leidžia nurodyti minimalų dydį geoobjekto, kurio užrašas rodomas, naudojant *Nepiešti užrašų geoobjektams, mažesniems nei*.
- Poligonų geoobjektams jūs taipogi galite filtruoti rodomus užrašus pagal tai, ar jie pilnai telpa į geoobjektą ar ne.
- Linijų geoobjektams jūs galite pasirinkti *Jungti sujungtas linijas, kad nebūtų užrašų dublių*, braižant gana erdvų žemėlapį kartu su *Atstumo* ar *Pakartojimo* parinktimis iš kortelės *Vieta*.

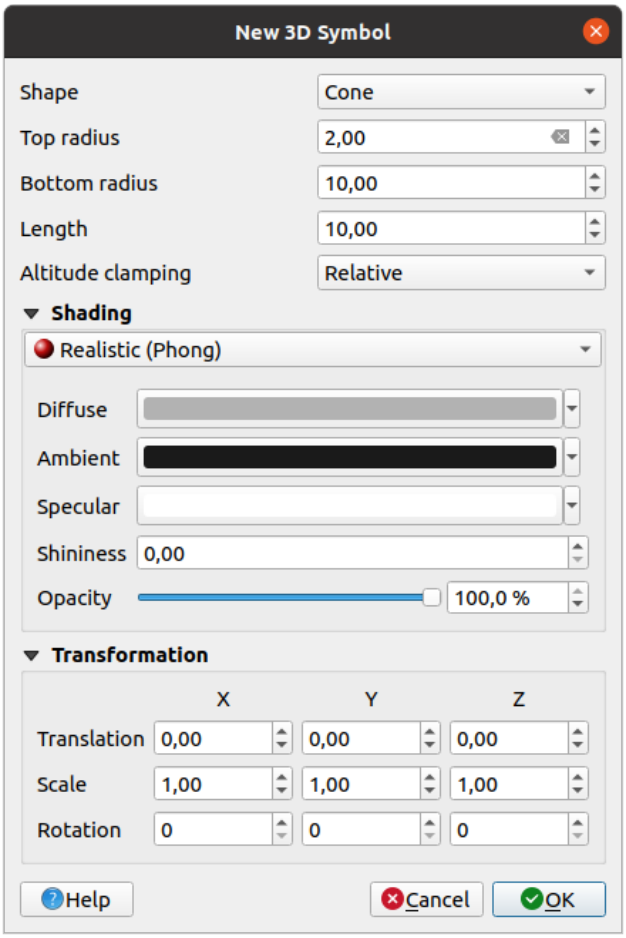
10.4 3D simbolių kūrimas

Stilių tvarkyklė padeda jums kurti ir saugoti visų geometrijų tipų 3D simbolius braižymui *3D žemėlapio vaizde*.

Kaip ir kitiems elementams, įjunkite kortelę *3D simboliai* ir išskleiskite meniu mygtuką , kad sukurtumėte:

- *3D taškų simbolius*
- *3D linijų simbolius*
- *3D poligonų simbolius*

10.4.1 Taškų sluoksniai

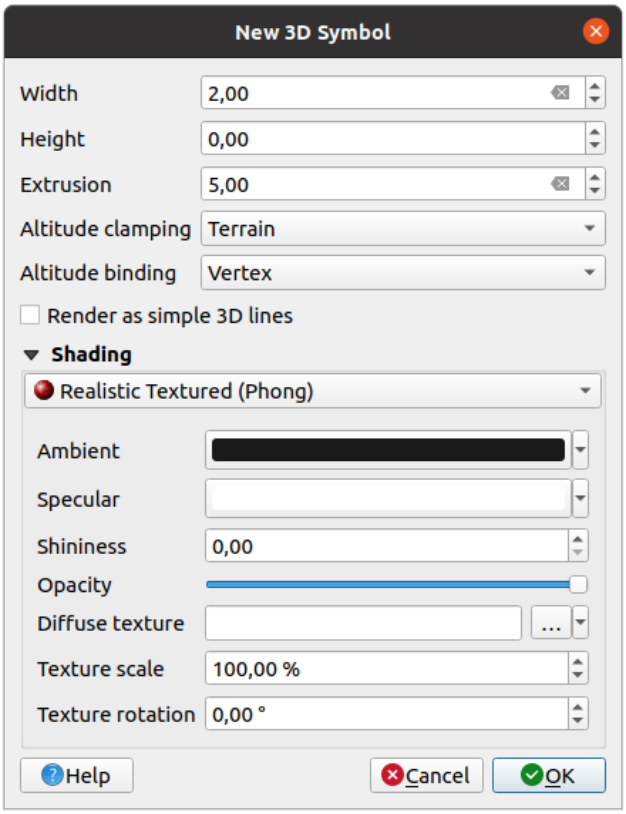


10.36 Fig. : 3D taško simbolio savybės

- Jūs galite nurodyti taškų simboliams naudoti skirtingas 3D *Formas*. Jie pagrinde apibrėžiami matmenimis, kurių matavimo vienetai atsižvelgia į projekto CRS. Galimi tipai yra:
  - *Sfera* apibrėžiama *Spinduliu*
  - *Cilindras* apibrėžiamas *Spinduliu* ir *Ilgiu*
  - *Kubas* apibrėžiamas *Dydžiu*
  - *Kūgis* apibrėžiamas *Viršutiniu spinduliu*, *Apatiniu spinduliu* ir *Ilgiu*
  - *Plokštuma* apibrėžiama *Dydžiu*
  - *Toras* apibrėžiamas *Spinduliu* ir *Mažuoju spinduliu*
  - *3D modelis*, naudojant 3D modelio failą: tarp palaikomų formatų yra wavefront `.obj`, `.glTF` ir `.fbx`. Modeliai gali būti failai diske, nutolę URL ar *įtraukti į projektą*. Bendruomenės sukurti simboliai bendrinami QGIS Hub.
  - *Stendas*, apibrėžiamas *Stendo aukščiu* ir *Stendo simboliu* (paprastai paremtu *žymeklio simboliu*). Simbolis bus stabilaus dydžio. Patogu vizualizuojant 3D taškų masyvų formas.
- *Aukščio prispaudimą* galima nurodyti *Absoliutų*, *Santykinį* ar *Paviršiaus*. Nustatymą *Absoliutus* galima naudoti, kai 3d vektorių aukščio reikšmės pateiktos kaip absoliutūs dydžiai nuo 0. *Santykinio* ir *Paviršiaus* atveju pateiktos reikšmės pridedamos prie atitinkamo paviršiaus aukščio.
- Galia nurodyti *šešėliavimo* savybes.

- Skiltyje *Transformacija* jūs galite simboliui taikyti paprastą transformaciją:
  - *Postūmis* pastums objektus x, y ir z ašyse.
  - *Mastelis* pakeis 3D formų dydį
  - *Pasukimas* aplink x-, y- ir z-ašis.

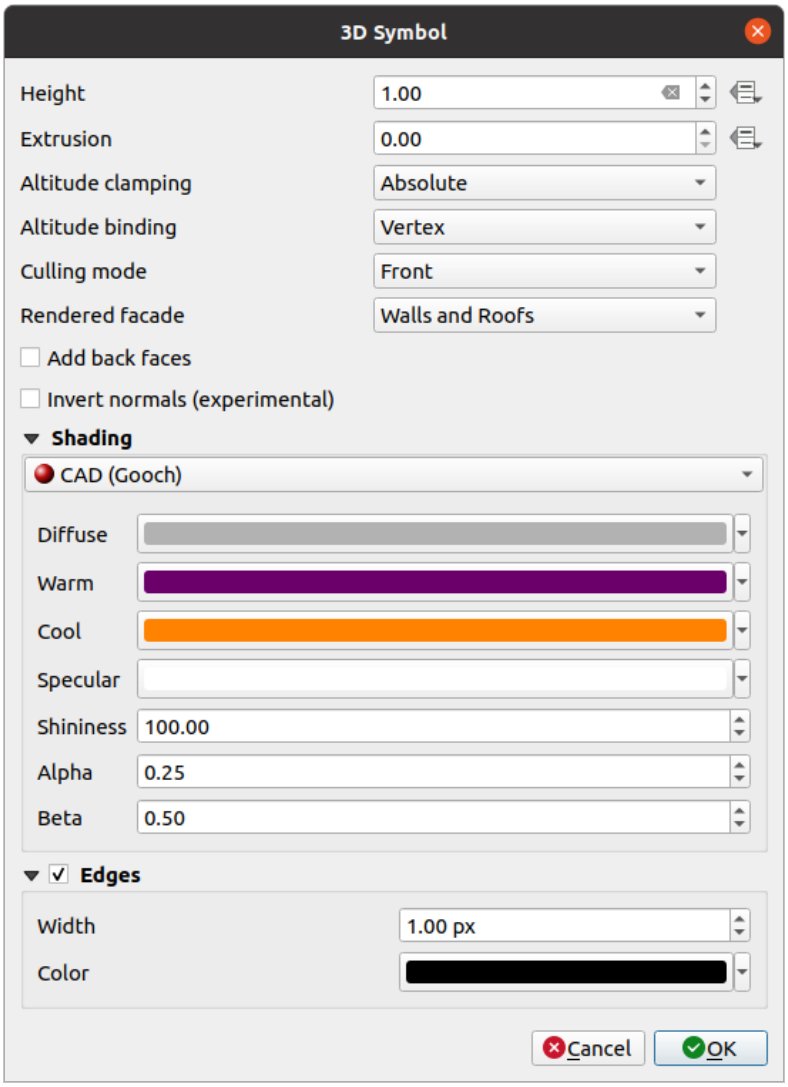
10.4.2 Linijų sluoksniai



10.37 Fig. : 3D linijos simbolio savybės

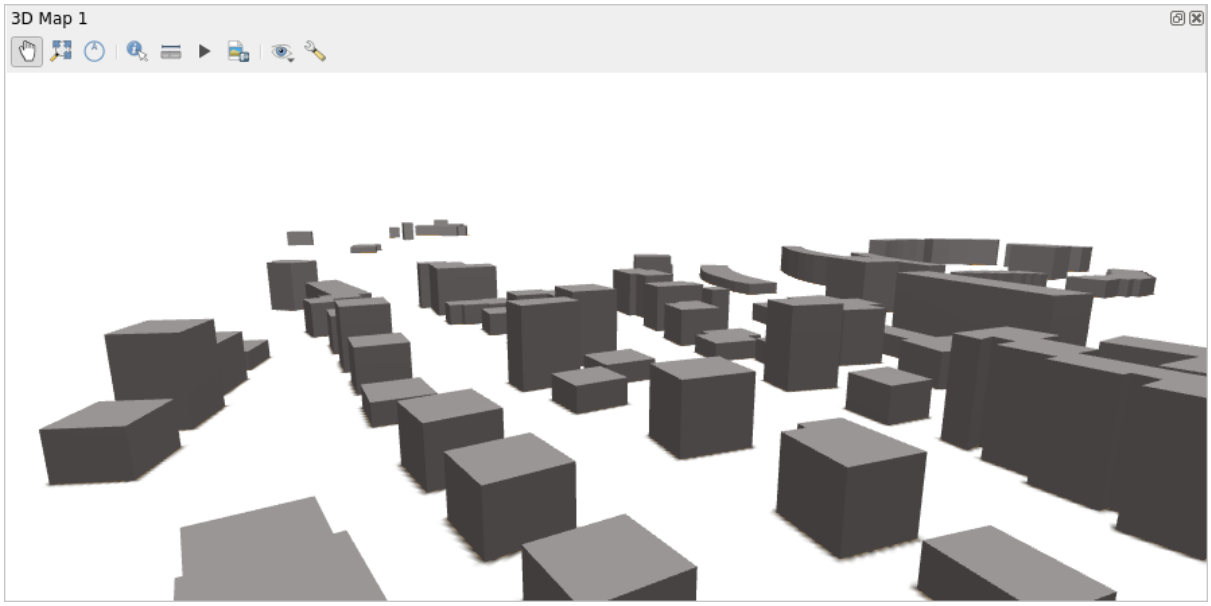
- Po *Pločio* ir *Aukščio* nustatymų jūs galite nurodyti vektorinių linijų *Išsikišimą*. Jei linijos neturi z-reikšmės, jūs šiuo nustatymu galite nurodyti 3d tūrius.
- Naudodami *Aukščio prispaudimą* jūs nurodo 3D linijų poziciją paviršiaus atžvilgiu, jei įtraukėte rastro aukščio duomenis ir kitus 3D vektorius.
- *Aukščio pririšimas* nurodo, kaip geoobjektai pritraukiami prie paviršiaus. Prie paviršiaus gali būti pritraukiama kiekviena geoobjekto *Viršūnė*, arba *Centroidas*.
- Galima *Braižyti kaip paprastas 3D linijas*.
- Galia nurodyti *šešėliavimo* savybes.

10.4.3 Poligonų sluoksniai



10.38 Fig. : 3D poligono simbolio savybės

- Kaip ir kitiems, *Aukštį* galima nurodyti CRS vienetais. Jūs taipogi galite naudoti mygtuką , kad permuštumėte reikšmę savo išraiška, kintamuoju arba atributų lentelės įrašu
- Vėlgi, trūkstant z-reikšmių galima naudoti *Išsikišimą*. Išsikišimui jūs taipogi galite naudoti mygtuką , jei norite naudoti vektorinio sluoksnio duomenis ir taip gauti skirtingų poligonų rezultatus:



10.39 Fig. : Nuo duomenų priklausantis išsikišimas

- Galima apibrėžti *Aukščio pritraukimą*, *Aukščio susiejimą*, kaip paaiškinta aukščiau.
- Simboliui taikomas *Išmetimo režimas* gali būti:
  - *Be išmetimo*: tai naudinga, kai reikia išvengti trūkstamų paviršių, kai polygonZ/multipatch duomenys neturi stabilaus viršūnių braižymo (pvz. visos pagal arba prieš laikrodžio rodyklę)
  - *Priekis*
  - ar *Nugara*
- *Braižomas fasadas* nurodo, kuriuos paviršius rodyti. Galimos reikšmės yra *Be fasadų*, *Sienos*, *Stogai* arba *Sienos ir stogai*
- ☑ *Pridėti fono piešinius*: kiekvienam trikampiui sukuria tiek priekinę, tiek galinę plokštumą su tinkamomis normalėmis - didesnių viršūnių duomenų kaina. Šią parinktį galima naudoti tvarkant šešėliavimo problemas (pvz. dėl duomenų su nepastovia viršūnių tvarka).
- ☑ *Invertuoti normales (eksperimentinis)*: gali būti naudingas taisant palei/prieš laikrodžio rodyklę einančių plokštumų viršūnių rikiuotes
- Galia nurodyti *šešėliavimo* savybes.
- Galima įjungti simbolių ☑ *Kraštų* rodymą ir nurodyti jiems *Plotį* ir *Spalvą*.

Patarimas

Geriausio 3D duomenų braižymo kombinacijos

Išmetimo režimas, *Pridėti fono plokštumas* ir *Invertuoti normales* - visi skirti pataisyti 3D duomenų išvaizdą, jei ji neatrodo gerai. Paprastai įkeliant kokius nors duomenis, geriausi iš pradžių pabandyti `išmetimo režimą=galas` ir `pridėti fono plokštumas=išjungta` - taip efektyviausia. Jei nubraižytas vaizdas atrodo negerai, pabandykite `pridėti fono plokštumas=įjungta` ir palikit `išmetimo režimą=be išmetimo`. Kitos kombinacijos yra sudėtingesnės ir naudingos tik kai kuriais scenarijais, priklausomai nuo to, kaip jūs sumaišėte įvesties duomenų aibę.

10.4.4 Tekstūros šešėliavimas

Šešėliavimas padeda išryškinti 3d objektų detales, kurios kitu atveju būtų paslėptos dėl scenos apšvietimo. Galų gale tai lengvesnė darbo medžiaga, nes jums nereikia rūpintis, kaip sukurti atitinkamą scenos apšvietimą, kad vizualizuotumėte geoobjektus.

QGIS naudoja įvairias šešėliavimo technikas, o jų prieinamumas priklauso nuo simbolio geometrijos tipo:

- *Realistiškas (Phong)*: aprašo kaip paviršius atspindi šviesą, kaip kombinaciją *Išsklaidyto* atspindžio nuo grubių paviršių su *Veidrodiniu* atspindžių nuo spindinčių paviršių (*Veidrodinis*). Ten pat yra *Aplinkos* parinktis, suteikianti nedidelį šviesos kiekį, kuris išsklaidomas po visą sceną. Naudokite slankiklį *Nepermatomumas*, kad braižytumėte pusiau permatomus 3d objektus. Daugiau informacijos rasite Phong atspindžio aprašyme.
- *Realistiškas pildymas (Phong)*: toks pat kaip ir *Realistinis (Phong)*, tik tiek, kad piešinys naudojamas kaip *Išsklaidymo tekstūra*. Šis piešinys gali būti diske, nutolusiame URL ar *įtrauktas į objektą*. *Tekstūros mastelis* ir *Tekstūros pasukimas* yra privalomi. Naudokite slankiklį *Nepermatomumas*, kad braižytumėte puriau permatomus 3D objektus.
- *CAD (Gooch)*: ši technika leidžia, kad šešėliai atsirastų tik vidutiniuose tonuose taip, kad kraštų linijos liktų vizualiai ryškesnės. Kartu su *Išsklaidymo*, *Veidrodiniu*, *Spindesio* parinktimis jums reikia pateikti ir *Šiltą* spalvą (paviršiams, kurie yra šviesos pusėje) ir *Vėsią* spalvą (tiems, kurie yra nukreipti nuo šviesos). Taipogi santykinę išsklaidytos šviesos įtaką vėsiai ir šiltai spalvoms, valdant atitinkamai *Alfa* ir *Beta* savybes. Daugiau informacijos apie Gooch šešėliavimą.
- *Metalo šiurkštumas*: fizika paremtas medžiagos braižymas, kuris tiksliai reprezentuoja kaip šviesą sąveikauja su paviršiais. Yra parinktys, leidžiančios nustatyti medžiagos *Bazinę spalvą*, *Metališkumą* ir *Šiurkštumą*.
- *Įtrauktos tekstūros* su 3D modelių forma

10.4.5 Taikymo pavyzdys

Kad suprastumėte aukščiau paaiškintus nustatymus, galite peržiūrėti https://app.merginmaps.com/projects/saber/luxembourg/tree.

SKYRIUS 11

Duomenų šaltinių valdymas

11.1 Duomenų atidarymas

Kaip dalis atviro kodo programinės įrangos ekosistemos, QGIS sukurta virš įvairių bibliotekų, kurios kartu su savo tiekėjais siūlo galimybes skaityti ir dažnai rašyti daugeliu formatų:

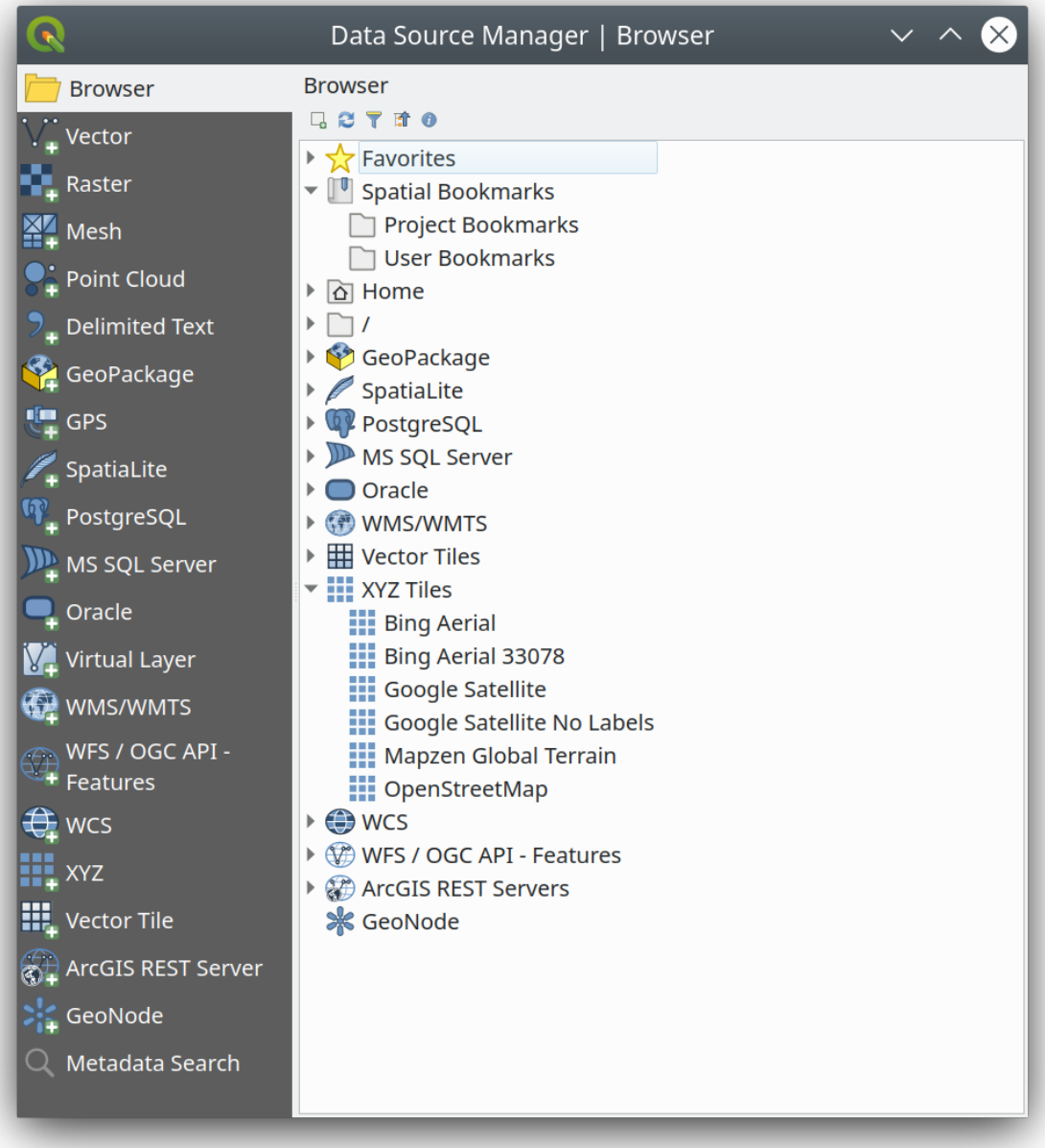
- Vektoriniai duomenų formatai, tai GeoPackage, GML, GeoJSON, GPX, KML, kableliais atskirtos reikšmės, ESRI formatai (Shapefile, Geodatabase…), MapInfo ir MicroStation failų formatai, AutoCAD DWG/DXF, GRASS ir daug kitų… Skaitykite pilną palaikomų vektorinių formatų sąrašą.
- Rastro duomenų formatai tai GeoTIFF, JPEG, ASCII tinklo XYZ, MBTiles, R ar Idrisi rastrai, GDAL virtualūs, SRTM, Sentinelio duomenys, ERDAS IMAGINE, ArcInfo Binary Grid, ArcInfo ASCII Grid ir daug kitų… Skaitykite pilną palaikomų rastro formatų sąrašą.
- Database formats include PostgreSQL, SQLite/SpatiaLite, Oracle, MS SQL Server, SAP HANA, MySQL…
- Žiniatinklio žemėlapiai ir duomenų paslaugos (WM(T)S, WFS, WCS, CSW, XYZ kaladėlės, ArcGIS paslaugos, …) taipogi palaikomos QGIS tiekėjų. Daugiau informacijos apie kai kuriuos iš jų rasite skyriuje *Darbas su OGC / ISO protokolais*.
- Jūs galite skaityti palaikomus failus iš archyvų aplankų ir naudoti QGIS vidinius formatus, tokius kaip QML failus (*QML - The QGIS Style File Format*) ir virtualius bei atminties sluoksnius.

GDAL ir vidiniai QGIS tiekėjai palaiko daugiau nei 80 vektorinių ir 140 rastro formatų.

> **Pastaba**
>
> Ne visi išvardinti formatai gali veiktu QGIS dėl įvairių priežasčių. Pavyzdžiui kai kuriems reikia išorinių nuosavybinių bibliotekų ar jūsų OS GDAL/OGR bibliotekų versijos gali nepalaikyti jums reikiamo formato. Norėdami pažiūrėti galimų formatų sąrašą, komandinėje eilutėje paleiskite `ogrinfo --formats` (vektoriniams) ir `gdalinfo --formats` (rastrams) arba žiūrėkite QGIS meniu punktą *Nustatymai* ► *Parinktys* ► *GDAL*.

Priklausomai nuo duomenų formato, QGIS yra keli įrankiai, leidžiantis atidaryti duomenų rinkinį, pagrinde juos galima rasti meniu:*Sluoksnis –> Pridėti sluoksnį –>* ar įrankinėje *Tvarkyti sluoksnius* (įjungiama per meniu *Vaizdas* ►

Įrankinės). Bet visi šie įrankiai veda į tą patį dialogą: *Duomenų šaltinių tvarkyklė*, kurį jūs galite atidaryti mygtuku
Atverti duomenų šaltinių tvarkyklę, o rasite jį *Duomenų šaltinių tvarkymo įrankinėje* ar spausdami `Ctrl+L`. Dialogas *Duomenų šaltinių tvarkyklė* (Fig. 11.1) teikia bendrą sąsają failais paremtiems duomenimis bei duomenų bazėms ar žiniatinklio paslaugoms, palaikomoms QGIS.



11.1 Fig. : QGIS duomenų šaltinių tvarkyklės dialogas

Be šio bendro pradžios taško, dar yra priedas *DB tvarkyklė*, kuris teikia galimybę analizuoti ir manipuliuoti prijungtas duomenų bazes. Daugiau informacijos apie DB tvarkyklės galimybes rasite skyriuje *DB Manager Plugin*.

Yra daug kitų įrankių, vidinių ar trečiųjų-šalių priedų, kurie padės jums atverti įvairius duomenų formatus.

Šiame skyriuje aprašomi tik pagal nutylėjimą QGIS esantys duomenų įkėlimo įrankiai. Pagrindinis dėmesys bus skiriamas dialogui *Duomenų šaltinių tvarkyklė*, bet be visų kortelių aprašymo taipogi bus tyrinėjami įrankiai pagal duomenų tiekėją ar formato specifiką.

11.1.1 Naršyklės skydelis

Naršyklė - tai vienas iš pagrindinių būdų greitai ir lengvai pridėti duomenis į jūsų projektus. Jis veikia kaip:

- *Duomenų šaltinių tvarkyklės* kortelė, įjungiama paspaudus mygtuką Atverti duomenų šaltinių tvarkyklę (`Ctrl+L`);
- QGIS skydelis, kurį jūs galite atverti per meniu *Vaizdas* ► *Skydeliai* (ar *Nustatymai* ► *Skydeliai*) arba paspaudę `Ctrl+2`.

Abiem atvejais *Naršyklė* padeda jums naršyti po jūsų failų sistemą ir tvarkyti geoduomenis, nepriklausomai nuo sluoksnio tipo (rastras, vektorius, lentelė) ar duomenų šaltinio formato (paprasti ar suspausti failai, duomenų bazės, žiniatinklio šaltiniai).

Naudotojo sąsajos tyrinėjimas

Naršyklės skydelio viršuje jūs rasite kelis mygtukus, kurie padės jums:

- Pridėti parinktus sluoksnius: jūs taipogi galite pridėti duomenis į žemėlapio drobę parinkdami sluoksnio kontekstiniame meniu **Pridėti parinktą sluoksnį(ius)**;
- Atnaujinti naršyklės medį;
- Filtruoti naršyklę konkrečių duomenų paieškai. Įveskite paieškos žodį ar šabloną ir naršyklė filtruos medį, kad rodytų tik kelius į atitinkamas DB lenteles, failų pavadinimus ar aplankus – kiti duomenys ir aplankai nebus rodomi. Žiūrėkite naršyklės skydelio(2) pavyzdį Fig. 11.2. Lyginimą galima daryti atsižvelgiant į raidžių dydį arba ne. Jį taipogi galima nustatyti į:
  - *Normalų*: rodyti elementus, kuriuose yra ieškomas tekstas
  - *Šablonas(ai)*: tinkinkite paiešką naudodami `?` ir/ar `*` simbolius nurodydami ieškomo teksto poziciją
  - *Reguliari išraiška*
- Sutraukti viską visą medį;
- Įjungti/išjungti savybių valdiklį: įjungus, naujas valdiklis pridedamas skydelio apačioje ir, kai galima, rodo parinkto elemento metaduomenis.

Skydelio *Naršyklė* įrašai išdėlioti pagal hierarchiją ir yra keli aukščiausio lygio įrašai:

1. *Parankiniai*, kur jūs galite padėti dažnai naudojamas vietas
2. *Erdvinės žymelės*, kur jūs galite laikyti dažnai naudojamas žemėlapio apimtis (žr. *Žemėlapio apimčių žymelės*)
3. *Projekto namai*: greitai prieigai prie aplanko, kuriame laikoma (dauguma) su projektų susijusių duomenų. Numatytoji reikšmė yra aplankas, kuriame yra projekto failas.
4. *Namų* aplankas failų sistemoje ir failų sistemos šakninis aplankas.
5. Prijungti vietiniai ar tinklo įrenginiai
6. Toliau išvardinti keli konteinerių / duomenų bazių tipai ir paslaugų protokolai, priklausomai nuo jūsų platformos ir turimų bibliotekų:
   - *GeoPackage*
   - *SpatiaLite*
   - *PostgreSQL*
   - *SAP HANA*
   - *MS SQL Serveris*

- *Oracle*
- *WMS/WMTS*
- *Vektorinės kaladėlės*
- *XYZ kaladėlės*
- *WCS*
- *WFS/OGC API-geoobjektai*
- *ArcGIS REST Serveris*

Sąveika su naršyklės elementais

Naršyklė palaiko pertempimą ir numetimą naršyklės viduje, iš naršyklės į drobę ir *Sluoksnių* skydelį bei iš *Sluoksnių* skydelio į sluoksnių konteinerius (pvz. GeoPackage) naršyklėje.

Projekto failo elementus naršyklėje galima išplėsti, rodant pilną projekto sluoksnių medį (įskaitant grupes). Projekto elementai veikia taip pat, kaip bet koks kitas naršyklės elementas, taigi juos galima tempti ir numesti naršyklės viduje (pavyzdžiui kopijuoti sluoksnio elementą į geopackage failą) ar pridėti į dabartinį projektą pertempiant ir numetant ar du kartus paspaudus.

Naršyklės skydelio elemento kontekstinis meniu atidaromas paspaudus ant jo dešinį pelės mygtuką.

Failų sistemos aplankų įrašams, kontekstinis meniu leidžia:

- *Naujas* ► sukurti parinktame elemente:
  - *Aplanką…*
  - *GeoPackage…*
  - *ShapeFile…*
- *Pridėti kaip parankinį*: parankinius aplankus galima bet kokiu metu pervadinti (*Pervadinti parankinį…*) ar išimti (*Išimti parankinį*).
- *Slėpti nuo naršyklės*: paslėptus failus galima pakeisti į matomus naudojant nustatymą *Nustatymai* ► *Parinktys* ► *Duomenų šaltiniai* ► *Paslėpti naršyklės keliai*
- *Greitai skenuoti šį aplanką*
- *Atverti aplanką*
- *Atverti terminale*
- *Savybės…*
- *Aplanko savybės…*

Vaikiniams įrašams, kurie yra projektų sluoksniai, kontekstiniame meniu bus palaikantys įrašai. Pavyzdžiui, ne duomenų bazės, ne vektorinių, rastro ar duomenų tinklelio paslaugų įrašai turės:

- *Eksportuoti sluoksnį* ► *Į failą…*
- *Pridėti sluoksnį į projektą*
- *Sluoksnio savybės*
- *Atverti su duomenų šaltinių tvarkykle…*
- *Tvarkyti* ► *Pervadinti „<name of file>“…* ar *Trinti „<name of file>“…*
- *Rodyti failuose*
- *Failo savybės*

Įraše *Sluoksnio savybės* jūs rasite (panašiai, kaip ir *vektorinių* ir *rastro* sluoksnių savybėse, kai sluoksniai pridedami į projektą):

- Sluoksnio *Metaduomenis*. Metaduomenų grupes: *Tiekėjo informacija* (jei įmanoma, *Kelias* bus hipernuoroda į šaltinį), *Identifikacija*, *Apimtis*, *Prieiga*, *Laukai* (vektoriniams sluoksniams), *Juostos* (rastro sluoksniams), *Kontaktai*, *Jungtys* (vektoriniams sluoksniams), *Nuorodos* (rastro sluoksniams), *Istorija*.
- *Peržiūros* skydelį
- Vektorinių šaltinių atributų lentelę (skydelyje *Atributai*).

Naudokite *Atverti su duomenų šaltinių tvarkykle…*, kad tiesiogiai atidarytumėte ir konfigūruotumėte duomenų šaltinį su *Duomenų šaltinių tvarkykle* naudojant jūsų duomenų šaltinio URI. Tai supaprastina sluoksnio pridėjimo iš *Naršyklės* procesą leidžiant jums nustatyti konkrečias duomenų šaltinio atidarymo parinktis. Šiuo metu tai veikia su vektoriniais (įskaitant tam skirtus GeoPackage įrašus), rastro ir SpatiaLite duomenų šaltiniais.

Norėdami pridėti į projektą sluoksnį naudojant *Naršyklę*:

1. Įjunkite *Naršyklę*, kaip aprašyta aukščiau. Bus parodytas jūsų failų sistemos, duomenų bazių ir žiniatinklio paslaugų medis. Jums gali prireikti prijungti duomenų bazes ir žiniatinklio paslaugas, kad jos pasirodytų (žr. tam skirtas skiltis).
2. Raskite sluoksnį sąraše.
3. Naudokite kontekstinį meniu, du kartus paspauskite pavadinimą arba pertempkite ir numeskite jį į *žemėlapio drobę*. Jūsų sluoksnis dabar pridėtas į *Sluoksnių skydelį* ir galite jį matyti žemėlapio drobėje.

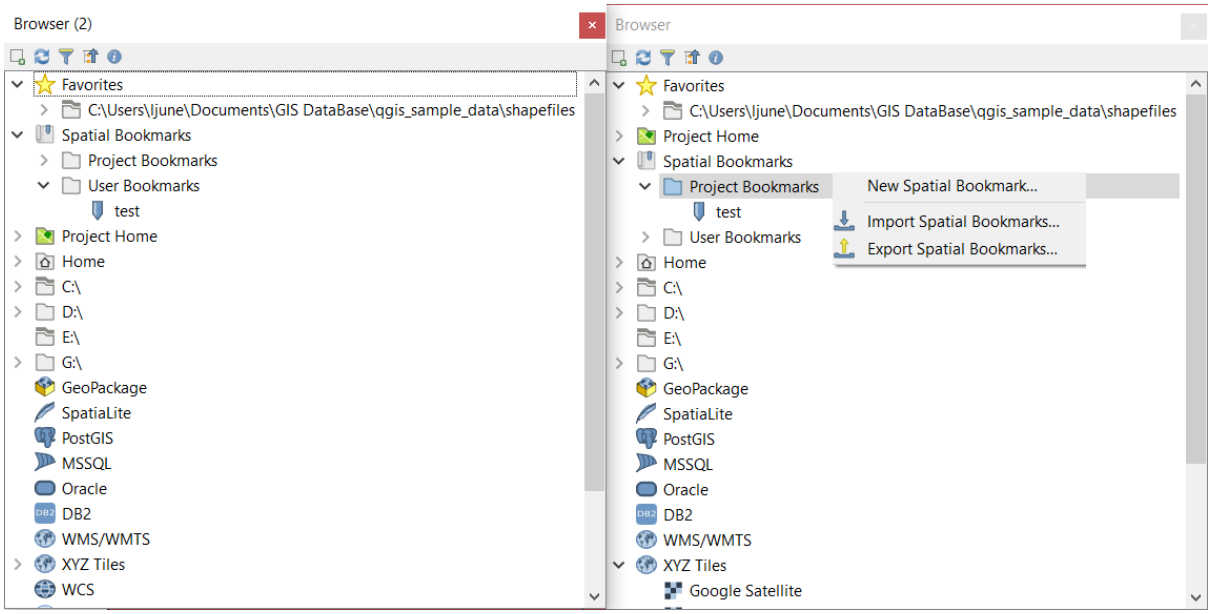
> **Patarimas**
>
> **Atverkite QGIS projektą tiesiai iš naršyklės**
>
> Jūs galite atverti QGIS projektą tiesiai iš Naršyklės skydelio du kartus paspaudę jo pavadinimą arba nutempę ir numetę jį į žemėlapio drobę.

Įkėlus failą jūs galite naršyti po jį naudodami žemėlapio navigacijos įrankius. Norėdami pakeisti sluoksnio stilių, atverkite dialogą *Sluoksnio savybės* du kartus paspaudę sluoksnio pavadinimą ar paspaudę dešinį pelės mygtuką ant pavadinimo legendoje ir kontekstiniame meniu parinkę *Savybės*. Daugiau informacijos apie vektorinių sluoksnių simbologijos nustatymus rasite skyriuje *Symbology Properties*.

Dešinys pelės paspaudimas naršyklės medyje padeda jums:

- failui ar lentelei - rodyti jo metaduomenis ar atverti jį jūsų projekte. Lenteles galima pervadinti, trinti ar ištuštinti.
- aplankui - jį pasidėti į parankinius ar paslėpti naršyklės medyje. Paslėptus aplankus galima valdyti kortelėje *Nustatymai* ► *Parinktys* ► *Duomenų šaltiniai*.
- valdyti jūsų *erdvines žymeles*: žymeles galima kurti, eksportuoti ir importuoti kaip `XML` failus.
- kurti jungtį su duomenų baze ar žiniatinklio paslauga.
- atnaujinti, pervadinti ar ištrinti schemą.

Jūs taipogi galite importuoti failus į duomenų bazes ar kopijuoti lenteles iš vienos schemos/duomenų bazės į kitą naudojant paprastą pertempimą ir numetimą. Galima naudoti antrą naršyklės skydelį, kad pertempimo metu nereikėtų toli sukti slankiklio. Tiesiog pažymėkite failą ir pertempkite bei numeskite jį į kitą skydelį.



11.2 Fig. : QGIS naršyklės skydeliai vienas greta kito

Patarimas

Pridėkite sluoksnius į QGIS tiesiog pertempdami juos iš jūsų OS failų naršyklės

Jūs taipogi galite pridėti failą(us) į projektą tempdami ir numesdami juos iš jūsų operacinės sistemos failų naršyklės į *Sluoksnių skydelį* ar žemėlapio drobę.

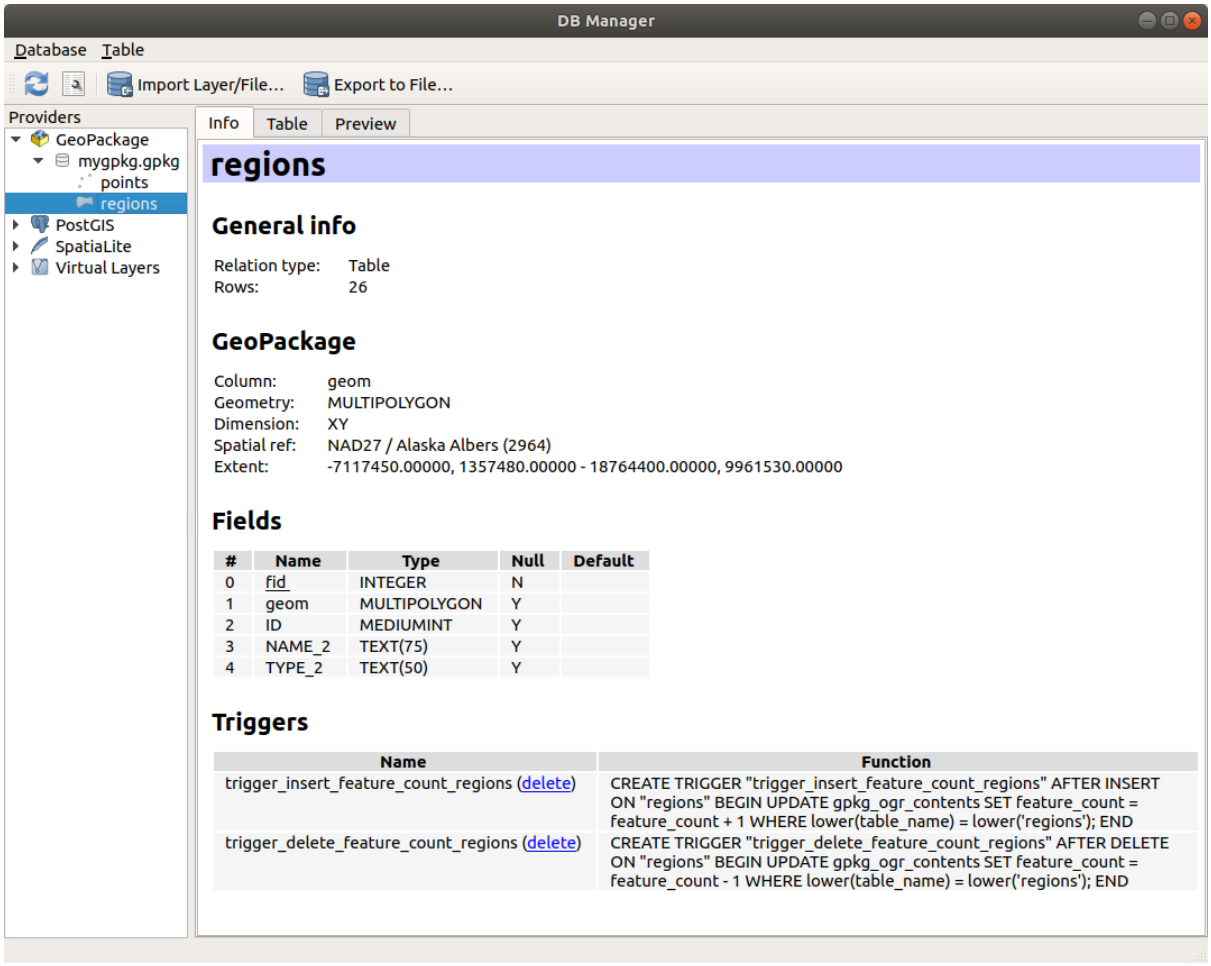
11.1.2 DB tvarkyklė

The *DB Manager* Plugin is another tool for integrating and managing spatial database formats supported by QGIS (PostgreSQL, SpatiaLite, GeoPackage, Oracle Spatial, MS SQL Server, Virtual layers). It can be activated from the *Plugins* ► *Manage and Install Plugins...* menu.

DB tvarkyklės priedas teikia kelias savybes:

- prisijungimas prie duomenų bazių ir jų struktūros bei turinio peržiūra
- duomenų bazių lentelių peržiūra
- sluoksnių pridėjimas į žemėlapio drobę dvigubu paspaudimu arba tempimu ir numetimu.
- sluoksnių pridėjimas į duomenų bazę iš QGIS naršyklės ar kitos duomenų bazės
- SQL užklausų kūrimas ir jų išvesties pridėjimas į žemėlapio drobę
- *virtualių sluoksnių* kūrimas

Daugiau informacijos apie DB tvarkyklės savybes galima rasti skyriuje *DB Manager Plugin*.



11.3 Fig. : DB tvarkyklės dialogas

11.1.3 Tiekėjų teikiami įkėlimo įrankiai

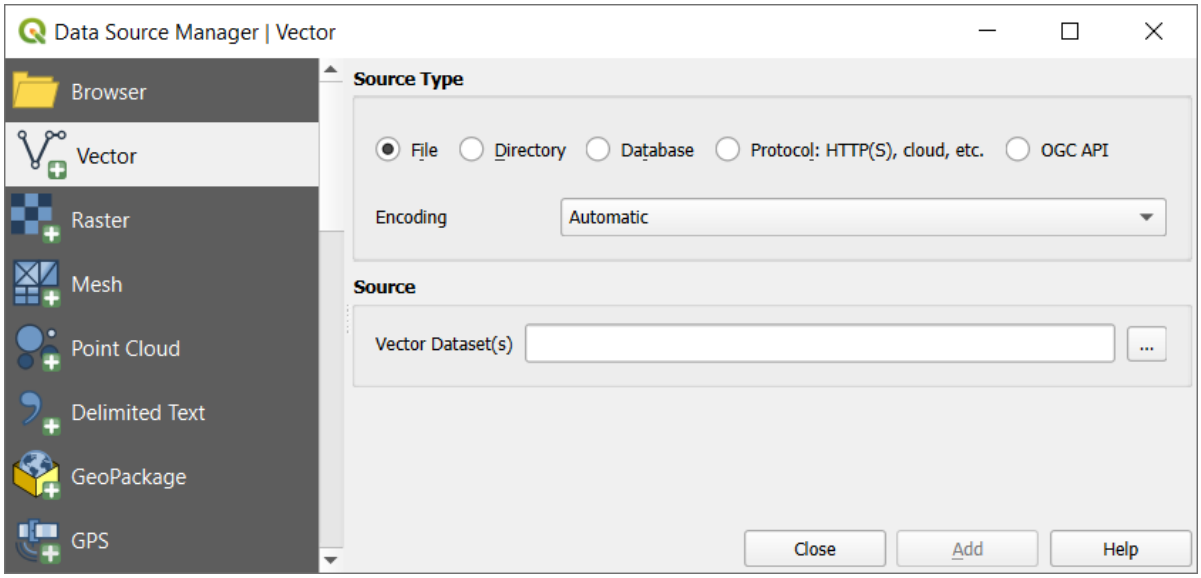
Be naršyklės skydelio ir DB tvarkyklės - pagrindinių QGIS teikiamų sluoksnių pridėjimo įrankių, jūs taipogi rasite įrankius, kurie yra naudojami tik su konkrečiais duomenų tiekėjais.

> **Pastaba**
>
> Kai kurie *išoriniai priedai* taipogi teikia įrankius, skirtus atidaryti QGIS konkrečių formatų failus.

Sluoksnio įkėlimas iš failo

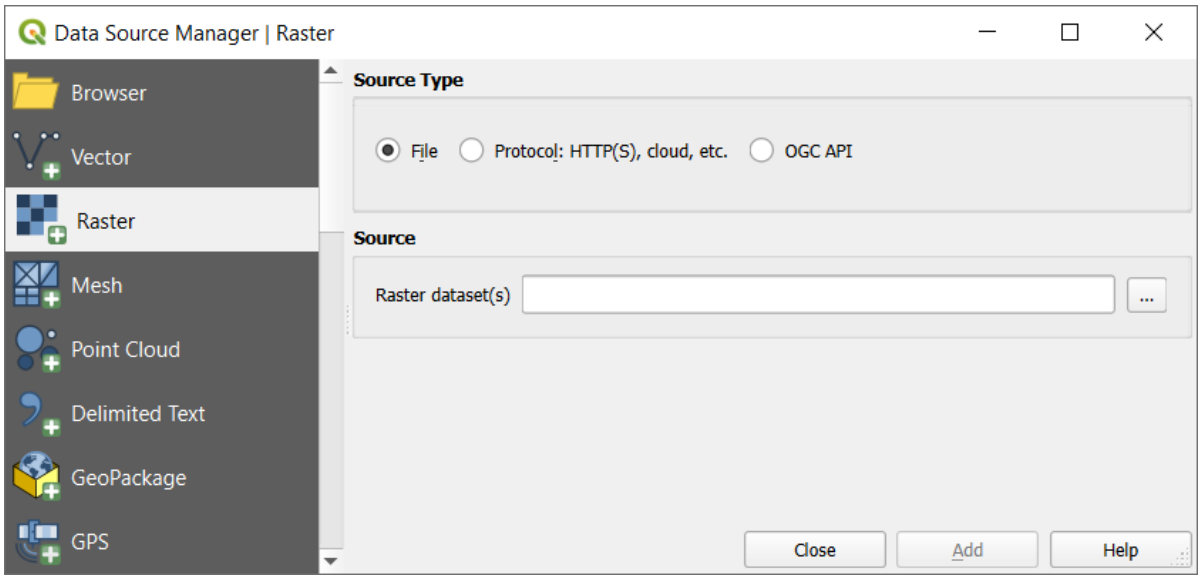
Norėdami įkelti sluoksnį iš failo:

1. Atidarykite sluoksnių tipo kortelę dialoge *Duomenų šaltinių tvarkyklė*, t.y. spauskite mygtuką Atverti duomenų šaltinių tvarkyklę (arba spauskite `Ctrl+L`) ir įjunkite paskirties kortelę, arba:
   - vektoriniams duomenims (tokiems kaip GML, ESRI Shapefile, Mapinfo ir DXF sluoksniams): spauskite `Ctrl+Shift+V`, parinkite meniu *Sluoksnis* ► *Pridėti sluoksnį* ► *Pridėti vektorinį sluoksnį* arba spauskite įrankinės mygtuką Pridėti vektorinį sluoksnį.



11.4 Fig. : Vektorinio sluoksnio pridėjimo dialogas

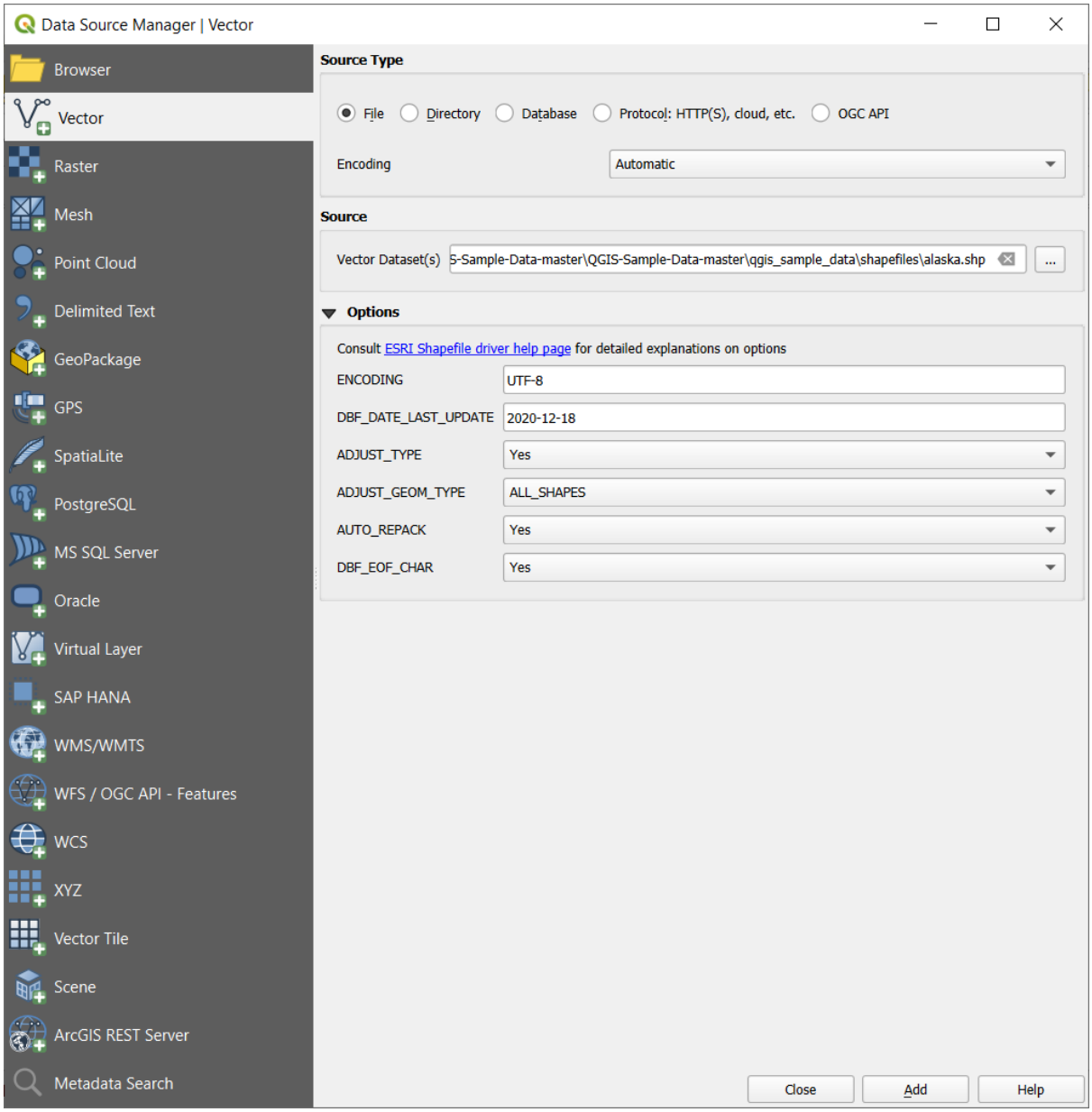
- rastro duomenims (tokiems kaip GeoTiff, MBTiles, GRIdded Binary ir DWG sluoksniams): spauskite `Ctrl+Shift+R`, parinkite meniu *Sluoksnis* ► *Pridėti sluoksnį* ► *Pridėti rastro sluoksnį* arba spauskite įrankinės mygtuką Pridėti rastro sluoksnį.



11.5 Fig. : Rastro sluoksnio pridėjimo dialogas

2. Parinkite šaltinio tipą *Failas*
3. Spauskite mygtuką … Naršyti
4. Naviguokite failų sistemoje ir įkelkite palaikomą duomenų šaltinį. Vienu metu galima įkelti daugiau nei vieną sluoksnį laikant `Ctrl` kai spaudžiama ant elementų dialoge arba laikant paspaustą `Shift`, kad parinktumėte diapazoną elementų paspaudus ant pirmo ir tada paskutinio diapazono elemento. Formatų filtre rodomi tik tie formatai, kurie buvo gerai išbandyti. Kitus formatus galima įkelti parinkus `Visi failai` (viršutinis iškrentančio meniu elementas).
5. Spauskite *Atidaryti*, kad įkeltumėte parinktą failą į *Duomenų šaltinių tvarkymo* dialogą.

Priklausomai nuo parinkto sluoksnio tipo, konfigūravimui bus prieinami papildomos *Parinktys* (koduotė, geometrijos tipas, lentelės filtravimas, failų rakinimas, duomenų formatavimas …). Šios parinktys detaliai aprašomos konkrečiuose GDAL vektorių ar rastro tvarkyklių dokumentacijose. Parinkčių viršuje esantis hypertekstas jus nuves tiesiai į parinkto failo formato atitinkamos tvarkyklės dokumentaciją.

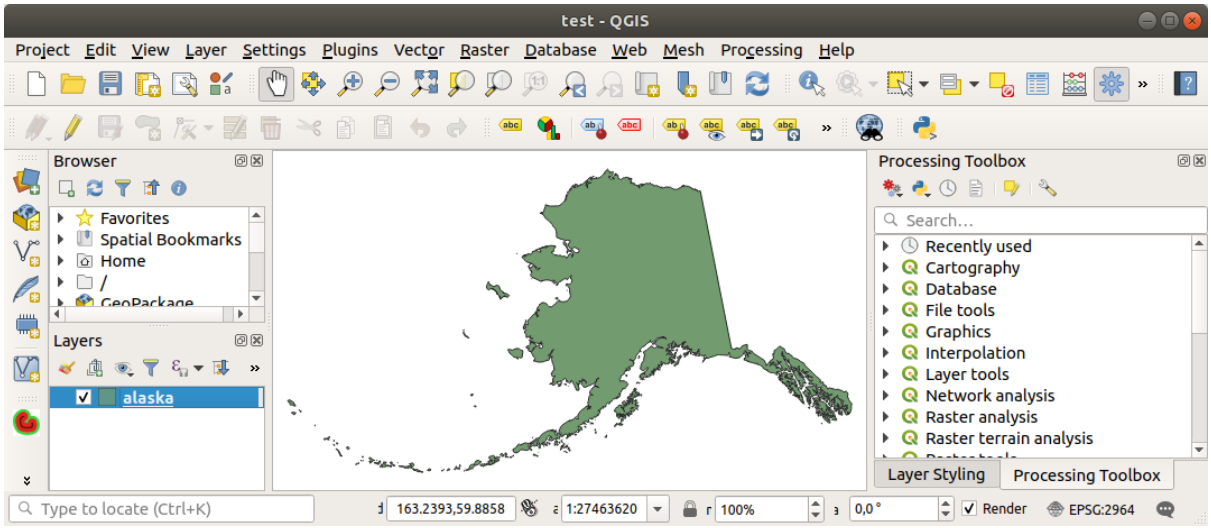


11.6 Fig. : Shape failo įkėlimas su atvertomis parinktimis

6. Spauskite *Pridėti*, kad įkeltumėte failą į QGIS ir parodytumėte jį žemėlapio vaizde. Kai pridedamos vektorinės duomenų aibės su keliais sluoksniais, bus rodomas dialogas *Parinkite pridedamus elementus*. Šiame dialoge jūs galite parinkti konkrečius jūsų duomenų rinkinio sluoksnius, kuriuos norite pridėti. Taipogi skiltyje *Parinktys* jūs galite pasirinkti:

   - ☑ *Pridėti sluoksnius į grupę*
   - ☑ *Rodyti sistemines ir vidines lenteles*
   - ☑ *Rodyti tuščius vektorinius sluoksnius*.

   Fig. 11.7 matomas QGIS po failo `alaska.shp` įkėlimo.



11.7 Fig. : QGIS su įkeltu Alaskos shape failu

Pastaba

Kadangi kai kurie formatai, tokie kaip MapInfo (pvz., `.tab`) ar Autocad (`.dxf`) leidžia viename faile kartu laikyti skirtingų tipų geometrijas, įkeliant tokius duomenų rinkinius atidaromas dialogas, kur reikia parinkti geometrijas kiekvienam sluoksniui.

Kortelės Pridėti vektorinį sluoksnį ir Pridėti rastro sluoksnį leidžia įkelti sluoksnius iš šaltini tipų, kitokių nei *Failas*:

- Jūs galite įkelti specifinius vektorinius formatus, tokius kaip `ArcInfo dvejetainis padengimas`, `UK. Nacionalinis perdavimo formatas` bei pradinį TIGER formatą iš `US Census Bureau` ar `OpenfileGDB`. Norėdami tai padaryti, parinkite *Aplanką* kaip *Šaltinio tipą*. Tokiu atveju, kai paspausite ... Naršyti, galėsite dialoge parinkti aplanką.

- Su šaltinio tipu *Duomenų bazė* jūs galite parinkti esamą duomenų bazės jungtį ar sukurti parinkto tipo naują. Kelios galimos duomenų bazės yra `ODBC`, `Esri Personal Geodatabase`, `MS SQL Server` kaip ir `PostgreSQL` ar `MySQL` .

  Pressing the *New* button opens the *Create a New OGR Database Connection* dialog whose parameters are among the ones you can find in *Įrašytos jungties kūrimas*. Pressing *Open* lets you select from the available tables.

- :guilabel:`Protokolas: HTTP(S), debesies ir panašiems šaltinių tipams atidaro duomenis, kurie saugomi vietoje arba tinkle, kur jie arba viešai prieinami, arba teikiami privačiuose komercinių debesų saugyklų paslaugomis. Palaikomi protokolų tipai yra:

  - `HTTP/HTTPS/FTP`, su *URI* ir, jei reikia, *autentifikacija*.
  - Debesų saugyklos, tokios kaip `AWS S3`, `Google Cloud Storage`, `Microsoft Azure Blob`, `Microsoft Azure Data Lake Storage`, `Alibaba OSS Cloud` ir `Open Stack Swift Storage`, palaiko tiesioginį valdymą per VSI *Prisijungimo parinktis*, kai pridedami OGR vektorių ar GDAL rastro sluoksniai. Jums reikia iš pradžių užpildyti *Kibirėlį ar konteinerį* ir *Objektų raktą*. Tada jūs galite pridėti reikiamą *Prisijungimo informaciją*.

    Pridedant OGR vektorinius ar GDAL rastro sluoksnius per debesimis paremtus protokolus, jūs taipogi galite nurodyti papildomas *Prisijungimo parinktis* konkrečiai tvarkyklei ar kibirėliui. Kai prisijungimo duomenys randami sluoksnio URI, jie nustatomi automatiškai. Tai leidžia skirtingiems sluoksniams naudoti skirtingus prisijungimo duomenis.

  - paslauga, palaikanti OGC `WFS 3` (vis dar eksperimentinis), naudojant `GeoJSON` ar `GEOJSON – atskirta naujomis eilutėmis` formatą ar paremtą `CouchDB` duomenų baze. Privaloma

nurodyti *URI*, galima nurodyti ir *autentifikaciją*.

– Visiems vektoriniams šaltinių tipams galima nurodyti *Koduotę* arba naudoti nustatymą *Automatinis* ►.

- Šaltinio tipas *OGC API* leidžia pasiekti vektorinius ir rastro duomenis iš serverių, kurie įgyvendina OGC API standartus. Norėdami naudoti šią parinktį:

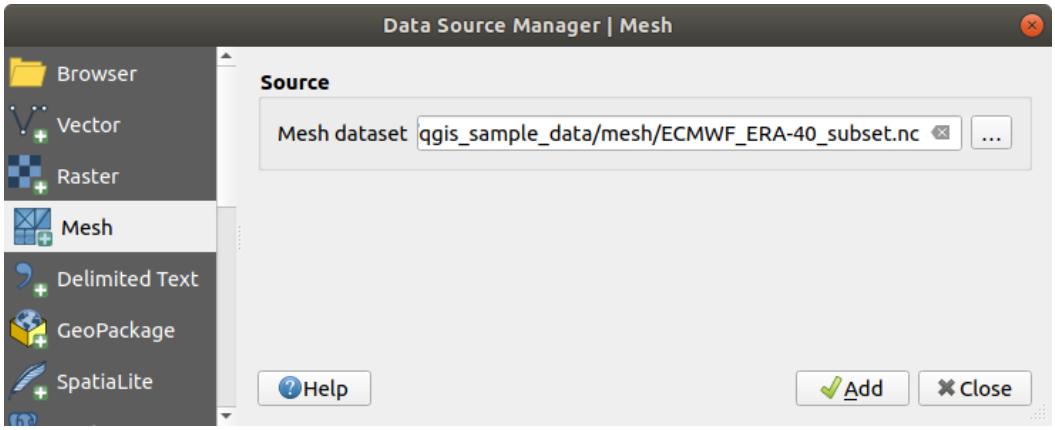
  1. Dialoge *Duomenų šaltinių tvarkyklė* parinkite *OGC API*.
  2. Įveskite OGC API paslaugos prieigos tašką, prie kurio norite prisijungti. Pastebėtina, kad nurodant prieigos tašką nereikia jam pridėti priešdėlio „OGCAPI:“.
  3. Spauskite *Prisijungti*, kad sukurtumėte jungtį su serveriu.

Tinklelio sluoksnio įkėlimas

Tinklelis yra nestruktūrinis tinklas, paprastai su laiko ar kitais komponentais. Erdviniame komponente yra aibė viršūnių, kraštinių ir plokštumų 2D ir 3D erdvėje. Daugiau informacijos apie tinklelio sluoksnius rasite skyriuje *Darbas su tinklelio duomenimis*.

Norėdami pridėti tinklelio sluoksnį į QGIS:

1. Atverkite dialogą *Duomenų šaltinių tvarkyklė* arba parenkant jį per meniu *Sluoksnis* ►, arba paspaudus mygtuką Atverti duomenų šaltinių tvarkyklę.
2. Kairiame skydelyje įjunkite kortelę *Tinklelis*
3. Spauskite mygtuką … Naršyti, kad parinktumėte failą. Palaikomi *Įvairūs formatai*.
4. Parinkite failą ir spauskite *Pridėti*. Sluoksnis bus pridėtas naudojant savą tinklelio braižymą.
5. Jei parinktame faile yra daug tinklelių sluoksnių, tada jums bus parodytas dialogas, kuriame jūsų paprašys pasirinkti įkeliamus posluoksnius. Pasirinkite ir spauskite *Gerai*, tada sluoksniai bus įkelti su savo tinklelio braižymu. Taipogi galima įkelti juos grupėje.

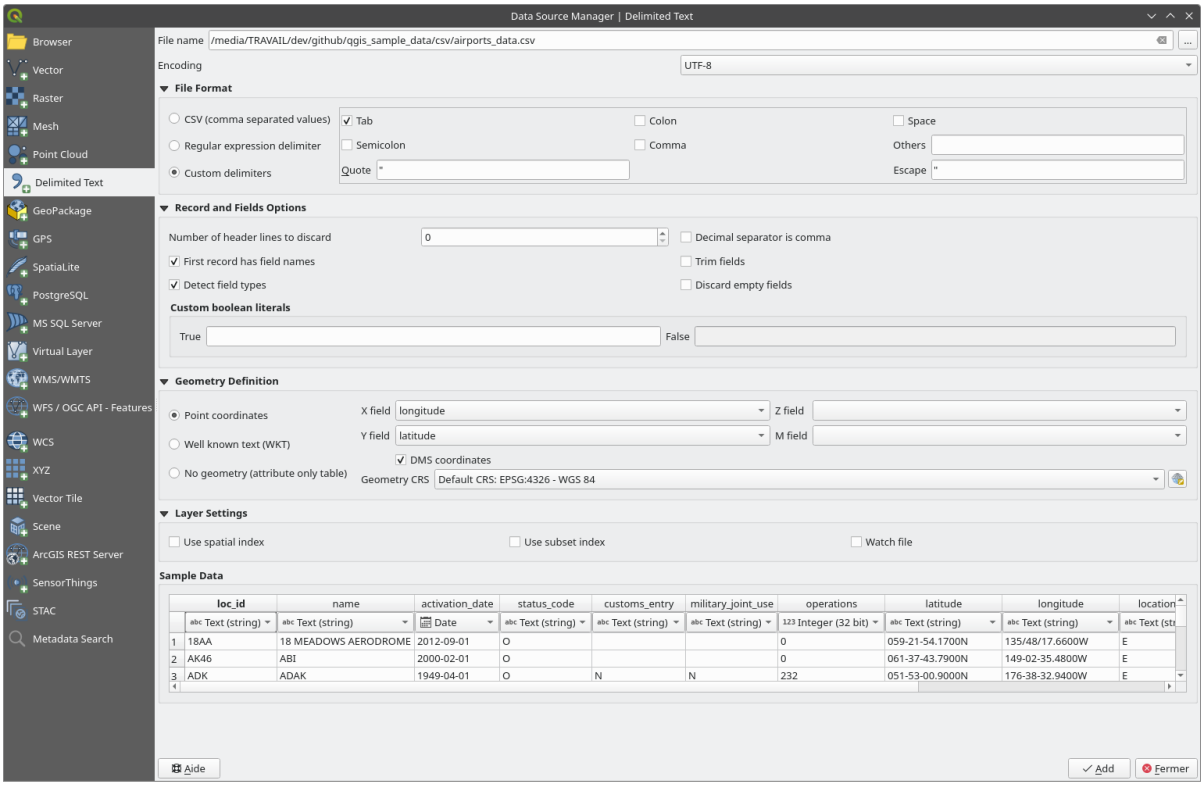


11.8 Fig. : Tinklelio kortelė Duomenų šaltinių tvarkyklėje

Atskirto tekstinio failo importavimas

Atskirus failus (pvz. `.txt`, `.csv`, `.dat`, `.wkt`) galima įkelti naudojant aukščiau aprašytus įrankius. Tokiu būdu jie bus rodomi kaip paprastos lentelės. Kartais atskirtuose teksto failuose gali būti koordinatės / geometrijos, kurias jūs galite norėti pavaizduoti. Tam ir sukurta galimybė *Pridėti atskirtą teksto sluoksnį*.

1. Spauskite piktogramą Atverti duomenų šaltinių tvarkyklę, kad atvertumėte *Duomenų šaltinių tvarkyklės* dialogą
2. Įjunkite kortelę *Atskirtas tekstas*
3. Parinkite importuojamą atskirto teksto failą (pvz., `qgis_sample_data/csv/elevp.csv`) paspausdami mygtuką ... Naršyti.
4. Sukonfigūruokite nustatymus, kad jie atitiktų jūsų duomenų aibės poreikius, kaip paaiškinta žemiau.



11.9 Fig. : Atskirto teksto dialogas

Failo formatas

Parinkus failą QGIS bando išnagrinėti failą su paskutiniu naudotu skirtuku, identifikuojant laukus ir eilutes. Kad QGIS galėtų teisingai išnagrinėti failą, labai svarbu parinkti teisingą skirtuką. Jūs galite nurodyti skirtuką pasirinkdami vieną iš:

- *CSV (kableliu atskirtos reikšmės)* kad naudotumėte kablelio simbolį.
- *Reguliarios išraiškos skirtukas* ir įveskite tekstą į lauką *Išraiška*. Pavyzdžiui jei norite pakeisti skirtuką į tabuliacijos simbolį, naudokite `\t` (tai reguliariose išraiškose naudojama kaip tabuliacijos simbolis).
- *Savi skirtukai*, pasirinkite iš iš anksto apibrėžtų skirtukų, tokių kaip `kablelis`, `tarpas`, `tabuliacija`, `kabliataškis`, ...

Įrašai ir laukai

Duomenų atpažinimui galima naudoti keletą kitų patogių parinkčių:

- *Skaičius išmetamų antraštės eilučių*: patogu, kai norite išvengti kelių pirmų failo eilučių importo metu, nes arba jos yra tuščios, arba jose yra kitoks formatavimas.
- *Pirmame įraše yra laukų pavadinimai*: pirmos eilutės reikšmės yra naudojamos laukų pavadinimams, kitu atveju QGIS naudoja laukų pavadinimus `field_1`, `field_2`...
- *Atpažinti laukų tipus*: automatiškai atpažįsta lauko tipą. Išjungus šią parinktį visi atributai laiko tekstiniais.
- *Trumpeninės dalies skirtukas yra kablelis*: galite priversti naudoti kablelį kaip trupmeninės dalies skirtuką.
- *Apkarpyti laukus*: leidžia jums apkarpyti laukus ir gale esančius tarpus.
- *Išmesti tuščius laukus*.
- *Savos loginės reikšmės*: leidžia jums pridėti savo eilučių porą, kuri bus aptinkama kaip loginės reikšmės.

Lauko tipo aptikimas

QGIS bando automatiškai aptikti laukų tipus (nebent varnelė *Aptikti laukų tipus* yra išjungta) tiriant papildomo CSVT failo turinį (žr. GeoCSV specifikacija) ir skenuojant visą failą, kad būtų įsitikinta, kad reikšmes tikrai galima konvertuoti be klaidų, atsarginis lauko tipas yra tekstas.

Aptiktas lauko tipas matosi po lauko pavadinimu pavyzdinių duomenų peržiūros lentelėje ir, jei reikia, jį galima pakeisti rankomis.

Palaikomi šie laukų tipai:

- `Loginis` nuo raidžių dydžio nepriklausančios tekstinės poros, kurios interpretuojamos kaip loginės reikšmės, yra `1`/`0`, `true`/`false`, `t`/`f`, `yes`/`no`
- `Sveikas skaičius (integer)`
- `Sveikas skaičius (integer - 64 bit)`
- `Dešimtainis skaičius`: dvigubo tikslumo slankaus kablelio skaičius
- `Data`
- `Laikas`
- `Data ir laikas`
- `Tekstas`

Geometrijos apibrėžimas

Išnagrinėjus failą, nustatykite *Geometrijos apibrėžimą* kaip

- *Taško koordinates* ir pateikite *X lauką*, *Y lauką*, *Z lauką* (3-matmenų duomenims) ir *M lauką* (matavimo matmeniui), jei sluoksnis yra taško geometrijos ir turi tokius laukus. Jei koordinatės nurodytos laipsniais/minutėmis/sekundėmis, įjunkite varnelę *DMS koordinatės*. Pateikite atitinkamą *Geometrijos CRS* naudojant valdiklį Parinkti CRS.
- *Gerai žinomo teksto (WKT)* parinktį, jei erdvinė informacija išreikšta kaip WKT: parinkite *Geometrijos lauką*, kuriame yra WKT geometrija ir parinkite atitinkamą *Geometrijos tipą* arba leiskite QGIS jį aptikti automatiškai. Pateikite atitinkamą *Geometrijos CRS* naudojant valdiklį Parinkti CRS.

- Jei faile yra neerdviniai duomenys, įjunkite *Nėra geometrijos (lentelėje tik atributai)* ir jis bus įkeltas kaip paprasta lentelė.

Sluoksnio nustatymai

Papildomai jūs galite įjungti:

- *Naudoti erdvinį indeksą*, kad pagerintumėte erdvinių geoobjektų rodymo ir parinkimo greitaveiką.
- *Naudoti poaibio indeksą*, kad pagerintumėte *poaibių filtrų* greitaveiką (kai jie apibrėžti sluoksnio savybėse).
- *Stebėti failą*, kad būtų stebima, kada failą keičia kitos programos kol veikia QGIS.

Tada spauskite *Pridėti*, kad pridėtumėte sluoksnį į žemėlapį. Mūsų pavyzdyje į projektą pridedamas taškų sluoksnis vardu `Elevation` ir jis elgiasi lygiai taip pat, kaip visi kiti QGIS žemėlapio sluoksniai. Šis sluoksnis yra šaltinio failo `.csv` užklausos rezultatas (taigi su juo susijęs) ir jį reikės *įrašyti*, jei norėsite gauti erdvinį sluoksnį diske.

Pavyzdiniai duomenys

Kai keičiate nagrinėjimo savybes, pavyzdiniai peržiūros duomenys keičiasi pagal jūsų nurodytus nustatymus.

Taipogi pavyzdinių duomenų lentelėje galima permušti automatiškai nustatytus stulpelių tipus.

DXF ar DWG failo importavimas

`DXF` ir `DWG` failus galima pridėti į QGIS tiesiog pertempus juos iš Naršyklės skydelio. Jūsų paprašys parinkti posluoksnius, kuriuos norite pridėti į projektą. Sluoksniai pridedami su atsitiktinėmis stiliaus savybėmis.

Pastaba

DXF failams su keliais geometrijos tipais (taškas, linija ir/arba poligonas), sluoksniai bus pavadinti kaip *<failas.dxf> esybės <geometry type>*.

Kad išlaikytumėte dxf/dwg failo struktūrą ir jo simbologiją QGIS'e, jūs galite naudoti tam skirtą įrankį *Projektas* ► *Importuoti/Eksportuoti* ► *Importuoti sluoksnius iš DWG/DXF…*, kuris leis jums:

1. importuoti elementus iš brėžinio failo į GeoPackage duomenų bazę.
2. pridėti importuotus elementus į projektą.

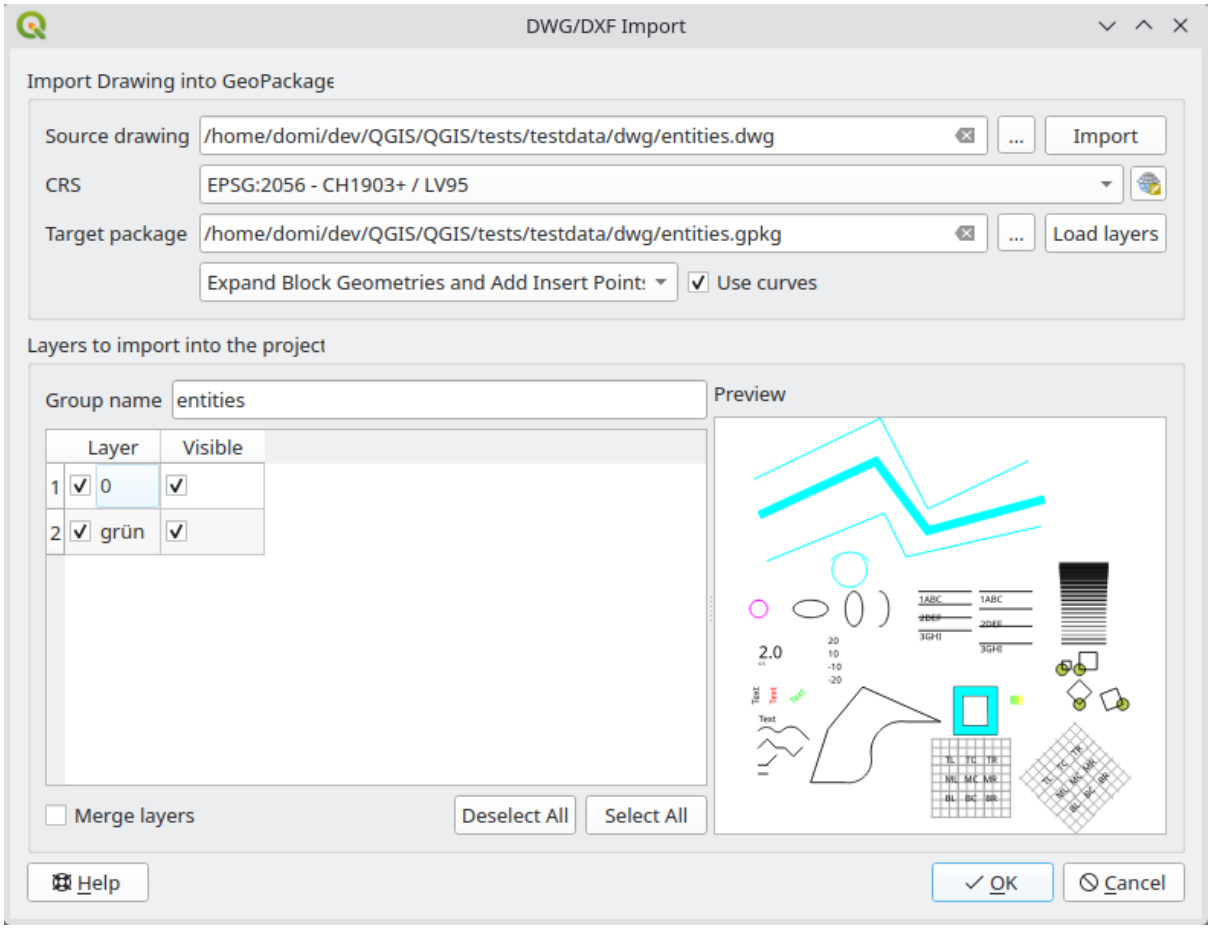
Dialoge *DWG/DXF importas*, norėdami importuoti brėžinio failo turinį:

1. Įveskite *Šaltinio brėžinio* vietą, t.y. DWG/DXF brėžinio failą, kurį reikia importuoti.
2. Nurodykite brėžinio failo duomenų koordinačių atskaitos sistemą.
3. Įveskite *Paskirties paketo* vietą, t.y. GeoPackage failą, į kurį bus įrašyti duomenys. Jei nurodomas esamas failas, jis bus perrašytas.
4. Parinkite, kaip importuoti `blokus`:
   - *Išskleisti blokų geometrijas*: importuoja brėžinio failo blokus kaip normalius elementus.
   - *Išskleisti bloko geometrijas ir pridėti įterpimo taškus*: importuoja brėžinio failo blokus kaip normalius elementus ir prideda įterpimo tašką kaip taškų sluoksnį.
   - *Tik pridėti taškus*: prideda blokų įterpimo tašką kaip taškų sluoksnį.
5. Įjunkite *Naudoti kreives*, kad keistumėte importuojamus sluoksnius į `kreivių` geometrijos tipą.

6. Naudokite mygtuką *Importuoti*, kad importuotumėte brėžinį į paskirties GeoPackage failą. GeoPackage duomenų bazė bus automatiškai užpildyta brėžinio failo turiniu. Priklausomai nuo failo dydžio, tai gali užtrukti.

Po to, kai `.dwg` ar `.dxf` duomenys importuojami į GeoPackage duomenų bazę, apatinėje dialogo dalyje esanti dalis užpildoma importuoto failo sluoksniais. Ten jūs galite pasirinkti ir pridėti sluoksnius į QGIS projektą:

1. Viršuje nurodykite *Grupės pavadinimą*, kad grupuotumėte brėžinių failus projekte. Pagal nutylėjimą ji nustatoma pagal šaltinio brėžinio failą.
2. Įjunkite rodomus sluoksnius: kiekvienas parinktas sluoksnis pridedamas į ad hoc grupę, kurioje yra brėžinio sluoksnio taškų, užrašų, linijų ir plotų geoobjektai. Sluoksnių stiliai panašūs į tuos, kurie buvo *CAD.
3. Parinkite ar sluoksnis turi būti matomas atidarant.
4. Įjungus parinktį *Sulieti sluoksnius*, visi sluoksniai bus vienoje grupėje.
5. Spauskite *OK*, kad atvertumėte sluoksnius QGIS.



11.10 Fig. : DWG/DXF failų importavimo dialogas

OpenStreetMap vektorių importavimas

OpenStreetMap projektas yra populiarus, nes daugumoje šalių negalima nemokamai gauti geoduomenų, tokių kaip kelių žemėlapiai. OSM projekto tikslas yra sukurti laisvą redaguojamą pasaulio žemėlapį naudojant GPS duomenis, oro fotografijas ir vietines žinias. Kad padėtų šiam tikslui, QGIS teikia OSM duomenų palaikymą.

Naudojant *Naršyklės skydelį* jūs galite įsikelti `.osm` failą į žemėlapio drobę, tokiu atveju jums bus parodytas dialogas, kuriame galėsite pasirinkti posluoksnius pagal geometrijos tipą. Įkeltuose sluoksniuose bus visi to geometrijos tipo duomenys iš `.osm` failo ir jie išlaikys `osm` failo duomenų struktūrą.

SpatiaLite sluoksniai

Kai pirmą kartą įkeliate duomenis iš SpatiaLite duomenų bazės, pradėkite:

- spausdami įrankinės mygtuką the Pridėti SpatiaLite sluoksnį
- parinkdami *Pridėti SpatiaLite sluoksnį…* iš meniu *Sluoksniai* ► *Pridėti sluoksnį*
- ar spausdami `Ctrl+Shift+L`

Tai pakels langą, kuris leis jums arba prisijungti prie QGIS jau žinomos duomenų bazės (kurią jūs pasirinktumėte iškrentančiame meniu) arba apibrėšite naują jungtį į naują duomenų bazę. Kad apibrėžtumėte naują jungtį, spauskite *Nauja* ir naudokite failų naršyklę, kad parodytumėte, kur yra jūsų SpatiaLite duomenų bazę, kuri yra failas su plėtiniu `.sqlite`.

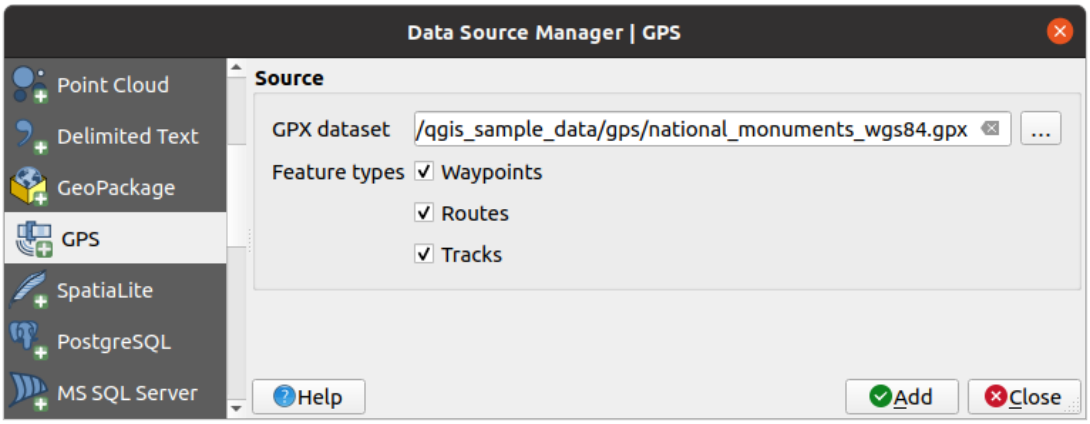
QGIS taipogi leidžia redaguojamus SpatiaLite vaizdus.

GPS

Yra daug failų formatų, leidžiančių laikyti GPS duomenis. QGIS naudoja formatą, vadinamą GPX (GPS eXchange format), kuris yra standartinis apsikeitimo formatas, kuriame gali būti bet koks skaičius taškų, maršrutų ir pėdsakų viename faile.

Naudokite mygtuką … Naršyti, kad parinktumėte GPX failą, tada naudokite varneles, kad parinktumėte geoobjektų tipus, kuriuos norite įkelti iš to GPS failo. Kiekvienas geoobjektų tipas bus įkeltas kaip atskiras sluoksnis.

Daugiau apie GPS duomenų apdorojimą rasite skyriuje *Working with GPS Data*.



11.11 Fig. : GPS duomenų įkėlimo dialogas

GRASS

Darbas su vektoriniais GRASS duomenimis aprašytas skyriuje *GRASS GIS Integration*.

Su duomenų baze susiję įrankiai

Įrašytos jungties kūrimas

Norint skaityti ir rašyti QGIS palaikomų duomenų bazių formatų lenteles, jums reikia sukurti jungti į tą duomenų bazę. Nors *QGIS naršyklės skydelis* yra paprasčiausias ir rekomenduojamas būdas prisijungti ir naudoti duomenų bazes, QGIS teikia ir kitus įrankius prisijungimui ir jų lentelių įkėlimui:

- *Add PostgreSQL Layer…* or by typing `Ctrl+Shift+D`
- *Pridėti Oracle Spatial sluoksnį…* ar paspaudus `Ctrl+Shift+O`
- *Pridėti MS SQL Server sluoksnį*
- *Pridėti SAP HANA Spatial sluoksnį…* ar paspaudus `Ctrl+Shift+G`

Šie įrankiai pasiekiami per *Sluoksnių tvarkymo įrankinę* ir meniu *Sluoksnis* ► *Pridėti sluoksnį* ►. Prisijungimas prie SpatiaLite duomenų bazės aprašytas skyriuje *SpatiaLite sluoksniai*.

Patarimas

Sukurkite jungtį į duomenų bazę QGIS naršyklės skydelyje

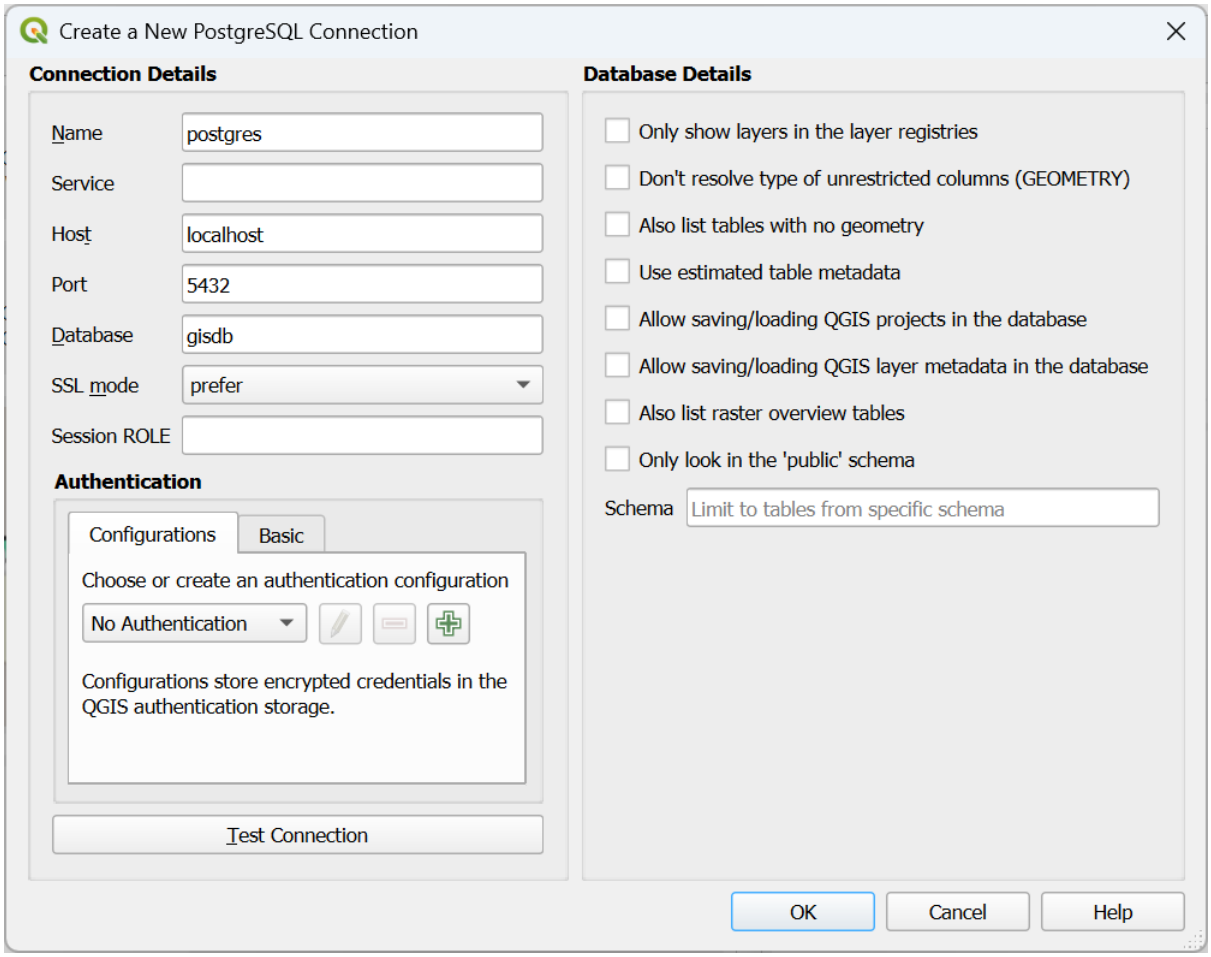
Parinkite naršyklės medyje atitinkamą duomenų bazės formatą, spauskite dešinį pelės mygtuką ir parinkite prisijungimą, tai pateiks jums prisijungimo prie duomenų bazės dialogą.

Dauguma prisijungimo dialogų turi bendrą struktūrą:

- skiltis su prisijungimo prie duomenų bazės informacija
- skiltis su parinktimis, nurodančiomis kurių duomenų galima prašyti iš duomenų bazės

Prisijungimas prie PostgreSQL

The first time you use a PostgreSQL data source, you must create a connection to a database that contains the data. Press the appropriate button as exposed above, opening the *PostgreSQL* tab of the *Data Source Manager* dialog. To access the connection manager, click on the *New* button to display the *Create a New PostgreSQL Connection* dialog.



11.12 Fig. : Create a New PostgreSQL Connection Dialog

- *Pavadinimas*: Šios jungties pavadinimas. Jis gali būti toks pat kaip *Duomenų bazė*.
- *Service*: Service parameter to be used alternatively to hostname/port (and potentially database). This can be defined in `pg_service.conf`. Check the *PostgreSQL Service connection file* section for more details.
- *Stotis*: Duomenų bazės mašinos pavadinimas. Jis turi galėti išsispręsti ir tikti atverti TCP/IP jungtį ar nusiųsti patikrinimo signalą į serverį. Jei duomenų bazė yra tame pačiame kompiuteryje kaip QGIS, tiesiog įveskite *localhost*.
- *Port*: Port number the PostgreSQL database server listens on. The default port for PostgreSQL is `5432`.
- *Duomenų bazė*: Duomenų bazės pavadinimas.
- *SSL režimas*: SSL šifravimo režimas. Galimos šios parinktys:
  - *Pirmenybė* (numatytoji): Man šifravimas nesvarbus, bet aš sutinku gaišti papildomą laiką šifravimui, jei serveris jį palaiko.
  - *Reikalauti*: Aš noriu, kad mano duomenys būtų užšifruoti ir aš sutinku su papildomai gaištamu laiku. Aš tikiu, kad tinklas užtikrins, kad aš visada prisijungiu prie to serverio, prie kurio noriu.
  - *Tikrinti CA*: Aš noriu, kad mano duomenys būtų šifruoti ir sutinku su papildomu reikiamu laiku. Aš noriu būti tikras, kad jungiuosi prie serverio, kuriuo pasitikiu.

  - *Tikrinti pilnai*: Aš noriu, kad mano duomenys būtų šifruoti ir sutinku su papildomu reikiamu laiku. Aš noriu būti tikras, kad jungiuosi prie serverio, kuriuo pasitikiu ir jis yra tas, kurį aš nurodžiau.
  - *Leisti*: Man saugumas nesvarbu, bet aš mokėsiu kainą už papildomą šifravimo darbą, jei serveris to reikalauja.
  - *Išjungti*: Man saugumas nesvarbus ir aš nenoriu gaišti papildomo laiko šifravimui.
- *Sesijos rolė*: naudojama nustatant dabartinės sesijos dabartinio naudotojo identifikatorių. Tai naudinga, kai reikia automatiškai duoti naujo objekto (lentelės, vaizdo, funkcijos) nuosavybę sesijos rolės grupei ir tokiu būdu dalinti susijusiomis teisėmis su visais sesijos rolės grupės nariais. Daugiau apie tai skaitykite <https://www.postgresql.org/docs/current/sql-set-role.html>`_.
- *Authentication*: For general details about the authentication dialog behavior, see *Autentifikacija*.

Pasirinktinai, priklausomai nuo duomenų bazės tipo, jūs galite įjungti šias varneles:

- ☑ *Rodyti tik sluoksnius sluoksnių registruose*
- ☑ *Nespręsti neribotų stulpelių tipų (GEOMETRY)*
- ☑ *Rodyti ir lenteles be geometrijos*: nurodo, kad pagal nutylėjimą reikia rodyti ir lenteles, kurios neturi geometrijos.
- ☑ *Naudoti įvertintus lentelės metaduomenis*: Inicializuojant sluoksnius, gali prireikti įvairių užklausų, kad būtų sužinotos duomenų bazės lentelėje saugomų geometrijų charakteristikos. Įjungus šią parinktį šios užklausos tiria tik dalį eilučių ir naudoja lentelės statistiką, o ne visą lentelę. Tai gali drastiškai pagreitinti veiksmus su dideliais duomenų rinkiniais, bet gali reikšti neteisingą sluoksnių charakteristiką (pvz. gali būti neteisingai nustatytas filtruotų sluoksnių geoobjektų skaičius) ir gali net sukelti keistą elgseną, jei stulpeliuose, kuriuose turėtų būti unikalios reikšmės, iš tikrųjų taip nėra.
- ☑ *Leisti įrašyti/įkelti QGIS projektus duomenų bazėje* - daugiau informacijos *čia*
- ☑ *Leisti įrašyti/įkelti QGIS sluoksnių metaduomenis duomenų bazėje* - daugiau informacijos *čia*
- ☑ *Also list raster overview tables*
- ☑ *Žiūrėti tik schemoje „public“*
- *Schema*: Leidžia nurodyti vieną schemą, kuria apsiriboti prisijungiant. Jei ji nurodyta, naršyklės skydelyje ir duomenų šaltinio pasirinkime jungtyje bus rodomos lentelės iš atitinkančios schemos. Tai galima naudoti ribojant duombazės darbą, kurio prireiks pildant jungties lenteles, kai kalba apie dideles duomenų bazių saugyklas.

Nustačius visus parametrus ir parinktis galite patikrinti jungtį paspausdami mygtuką *Bandyti jungtį* arba pritaikyti spausdami mygtuką *Gerai*.

PostgreSQL Service connection file

Paslaugos jungties failas leidžia PostgreSQL jungties parametrus susieti su vienu paslaugos pavadinimu. Tokį paslaugos pavadinimą gali būti nurodyti klientas ir bus naudojami susiję nustatymai.

Jis vadinasi `.pg_service.conf` *nix sistemose (GNU/Linux, macOS etc.) ir `pg_service.conf` Windows.

Paslaugos failas gali atrodyti taip:

```
[water_service]
host=192.168.0.45
port=5433
dbname=gisdb
user=paul
password=paulspass

[wastewater_service]
host=dbserver.com
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
dbname=water
user=waterpass
```

Pastaba

Aukščiau pateiktame pavyzdyje yra dvi paslaugos: `water_service` ir `wastewater_service`. Jūs galite jas naudodami prisijungdami iš QGIS, pgAdmin ir t.t. nurodydami tik paslaugos, prie kurios norite prisijungti, pavadinimą (be skliaustelių). Jei norite naudoti paslaugą su `psql`, galite tai padaryti taip `psql service=water_service`.

Visus PostgreSQL parametrus galite rasti čia

Pastaba

Jei nenorite įrašyti slaptažodžių į paslaugos failą, taipogi galite naudoti variantą .pg_pass.

Pastaba

QGIS Serveris ir paslauga

Kai naudojamas paslaugų failas ir QGIS Serveris, jūs privalote sukonfigūruoti paslaugą ir serverio pusėje. Galite laikytis instrukcijų, pateiktų QGIS Serverio dokumentacijoje.

*nix operacinėse sistemose (GNU/Linux, macOS etc.) jūs galite įrašyti `.pg_service.conf` failą į naudotojų namų aplanką ir PostgreSQL klientai į jį automatiškai atsižvelgs. Pavyzdžiui, jei prisijungęs naudotojas yra `web`, `.pg_service.conf` reikia įrašyti į aplanką `/home/web/`, kad jis veiktų (nenurodant jokių aplinkos kintamųjų).

Jūs galite nurodyti paslaugos failo vietą sukurdami aplinkos kintamąjį `PGSERVICEFILE` (pvz. įvykdykite komandą `export PGSERVICEFILE=/home/web/.pg_service.conf` jūsų *nix OS, kad laikinai nustatytumėte kintamąjį `PGSERVICEFILE`).

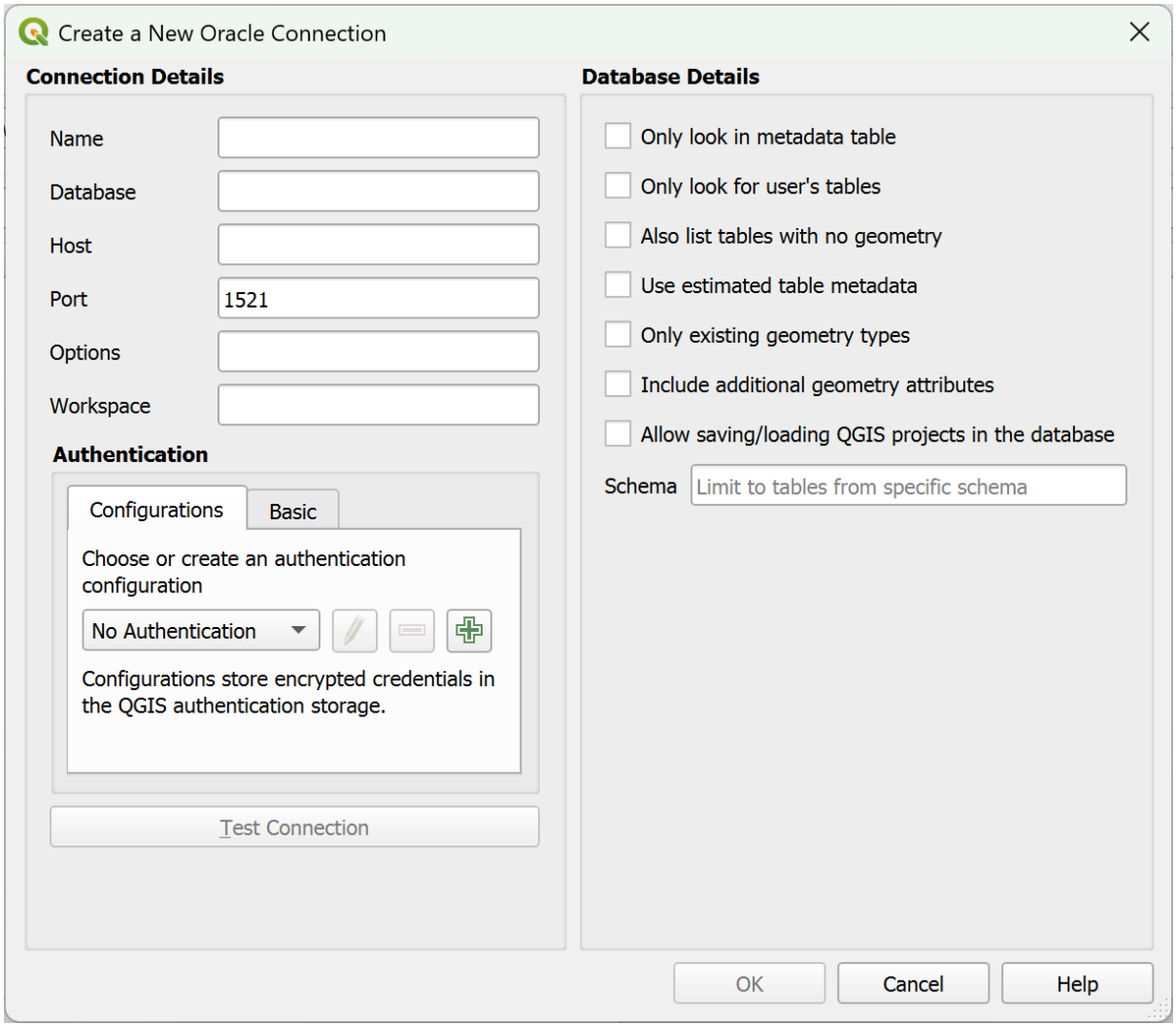
Jūs taipogi galite padaryti paslaugų failą prieinamu visai sistemai (visiems naudotojams) arba padėdami `.pg_service.conf` failą į `pg_config --sysconfdir` arba pridėdami aplinkos kintamąjį `PGSYSCONFDIR`, kad nurodytumėte aplanką, kuriame yra paslaugos failas. Jei paslauga su tokiu pačiu pavadinimu apibrėžta ir naudotojo, ir sistemos faile, bus naudojamas naudotojo failas.

Įspėjimas

Naudojant Windows yra keli įspėjimai:

- Paslaugos failą reikia įrašyti kaip `pg_service.conf`, o ne `.pg_service.conf`.
- Paslaugos failą reikia įrašyti Unix formatu, kad jis veiktų. Tai galima padaryti pavyzdžiui atidarius jį su Notepad++ ir parinkus *Edit* ► *EOL Conversion* ► *UNIX Format* ► *File save*.
- Jūs galite pridėti aplinkos kintamuosius keliais būdais. Patikrintas būdas, kuris užtikrintai veikia, yra *Control Panel* ► *System and Security* ► *System* ► *Advanced system settings* ► *Environment Variables* pridėti `PGSERVICEFILE` su keliu - pvz. `C:\Users\John\pg_service.conf`
- Pridėjus aplinkos kintamąjį jums gali tekti iš naujo paleisti jūsų kompiuterį.

Prisijungimas prie Oracle Spatial



11.13 Fig. : Create a New Oracle Connection Dialog

Oracle Spatial erdvinės savybės padeda naudotojams valdyti geografinius ir vietos duomenis naudojant savus tipus Oracle duomenų bazėje. Šis jungties dialogas siūlo:

- *Name*: A name for this connection. It can be the same as *Database*;
- *Database*: SID or SERVICE_NAME of the Oracle instance;
- *Host*: The name of the database host;
- *Port*: Port number the Oracle database server listens on. The default port is `1521`;
- *Options*: Oracle connection specific options (e.g. OCI_ATTR_PREFETCH_ROWS, OCI_ATTR_PREFETCH_MEMORY). The format of the options string is a semicolon separated list of option names or option=value pairs;
- *Workspace*: Workspace to switch to;
- *Authentication*: For general details about the authentication dialog behavior, see *Autentifikacija*.

Pasirinktinai galite įjungti šias varneles:

- *Ieškoti tik metaduomenų lentelėje*: rodo tik tas lenteles, kurios yra vaizde `all_sdo_geom_metadata`. Tai gali pagreitinti pradinį erdvinių lentelių parodymą.

- ☑ *Ieškoti tik naudotojo lentelėse*: ieškant erdvinių lentelių apsiribojama tik lentelėmis, kurių savininkas yra dabartinis naudotojas.
- ☑ *Rodyti ir lenteles be geometrijos*: nurodo, kad pagal nutylėjimą reikia rodyti ir lenteles, kurios neturi geometrijos.
- ☑ *Naudoti įvertintas lentelių statistikas sluoksnių metaduomenims*: kai ruošiamas sluoksnis, reikia turėti įvairius Oracle lentelės metaduomenis, tokius kaip lentelės įrašų skaičius, geometrijos tipas, geometrijos stulpelių duomenų erdvinė apimtis. Jei lentelėje yra daug įrašų, šių rodiklių nustatymas gali užtrukti daug laiko. Įjungus šią parinktį bus vykdomi šie greiti metaduomenų nustatymo veiksmai: įrašų skaičius nustatomas pagal `all_tables.num_rows`. Lentelės apimtis visada nustatoma naudojant funkciją SDO_TUNE.EXTENTS_OF, net jei taikomas sluoksnio filtras. Lentelės geometrija nustatoma pagal pirmus 100 ne null geometrijos įrašų.
- ☑ *Tik esami geometrijos tipai*: rodo tik esamos geometrijos tipus ir nesiūlo pridėti kitų.
- ☑ *Įtraukti papildomus geometrijos atributus.*
- ☑ *Leisti įrašyti/įkelti QGIS projektus duomenų bazėje* - daugiau informacijos *čia*
- *Schema*: Leidžia nurodyti vieną schemą, kuria apsiriboti prisijungiant. Jei ji nurodyta, naršyklės skydelyje ir duomenų šaltinio pasirinkime jungtyje bus rodomos lentelės iš atitinkančios schemos. Tai galima naudoti ribojant duombazės darbą, kurio prireiks pildant jungties lenteles, kai kalba apie dideles duomenų bazių saugyklas.

Patarimas

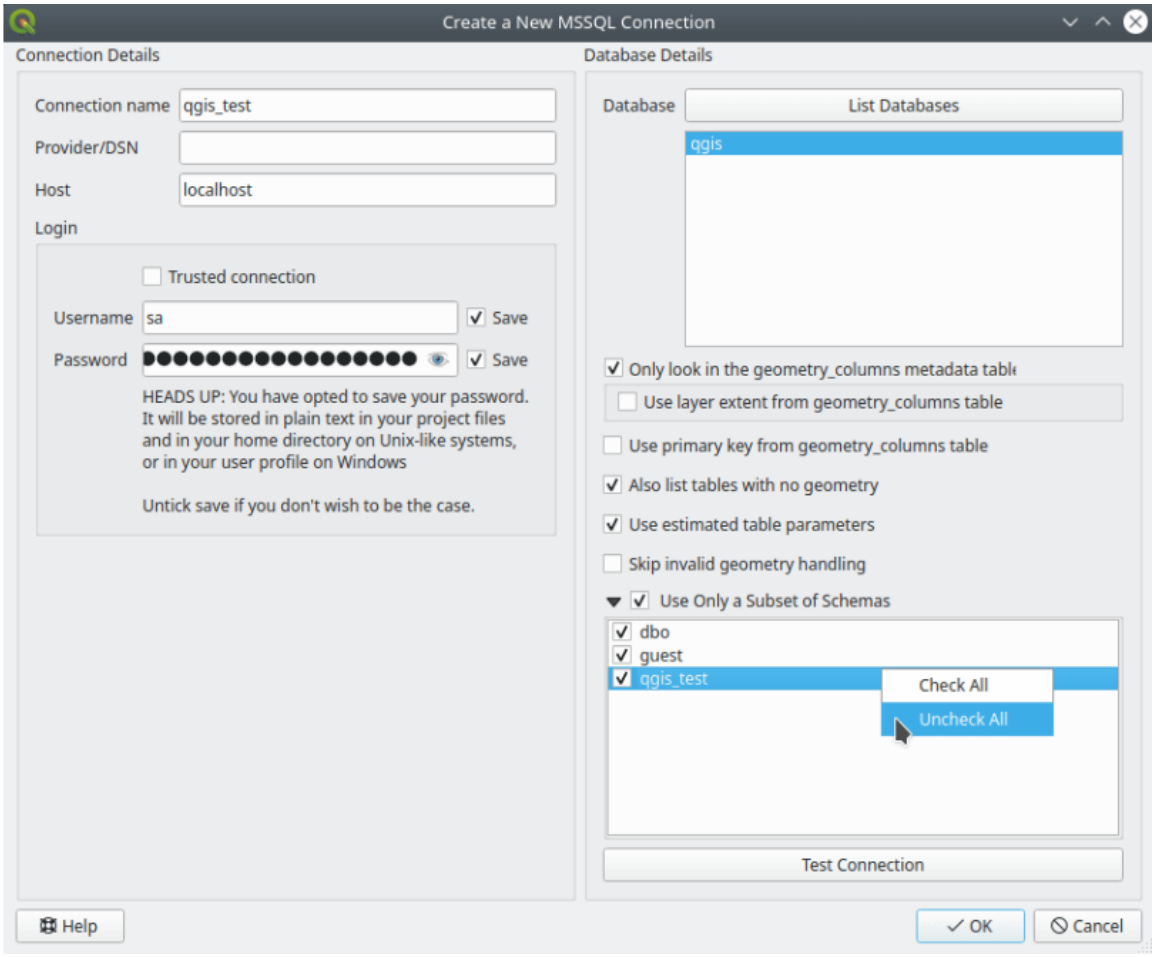
Oracle Spatial sluoksniai

Paprastai Oracle Spatial sluoksnis apibrėžiamas įrašų lentelėje **USER_SDO_METADATA**.

Kad įsitikintumėte, kad parinkimo įrankiai veikia korektiškai, rekomenduojama, kad jūsų lentelės turėtų **pirminį raktą**.

Prisijungimas prie MS SQL Server

Kaip paminėta *Įrašytos jungties kūrimas*, QGIS leidžia jums sukurti MS SQL Server jungtį naudojant *Duomenų šaltinių tvarkyklę.*



11.14 Fig. : MS SQL Server jungtis

Norėdami sukurti naują MS SQL Server jungtį jūs turite pateikti toliau išvardintą informaciją *Jungties detalių* dialoge:

- *Jungties pavadinimas*
- *Tiekėjas/DSN*
- *Mašina*
- *Prisijungimo* informacija. Jūs galite pasirinkti *Įrašyti* jūsų prisijungimo duomenis.

Nueikite į skiltį *Duomenų bazės detalės* ir spauskite mygtuką *Duomenų bazių sąrašas*, kad peržiūrėtumėte pasiekiamus duomenų rinkinius. Parinkite duomenų rinkinius, kurių jums reikia, tada spauskite *Gerai*. Pasirinktinai jūs galite vykdyti *Bandyti jungtį*. Paspaudus *Gerai*, dialogas *Kurti naują MS SQL Server jungtį* užsidarys, *Duomenų šaltinių tvarkyklėje* spauskite *Prisijungti*, parinkite sluoksnį ir spauskite *Pridėti*.

Pasirinktinai galite įjungti šias parinktis:

- *Ieškoti tik metaduomenų lentelėje geometry_columns*: skenuojant rodo tik tas lentas, kurios yra metaduomenų lentelėje `geometry_columns`. Tai gali pagreitinti lentelių skenavimą.
- *Naudoti sluoksnio apimtį iš lentelės geometry_columns*: ši parinktis, priklausanti nuo ankstesnės, leidžia QGIS praleisti apimties skaičiavimą įkeliant sluoksnius, taip sumažinant įkrovimui reikiamą laiką. Remiamasi rankiniu būdu nurodyta apimtimi naudojant QGIS-specifinius laukus (`qgis_xmin`, `qgis_xmax`, `qgis_ymin`, `qgis_ymax`) lentelėje `geometry_columns`.
- *Naudoti pirminį raktą iš lentelės geometry_columns*: leidžia QGIS praleisti pirminio rakto skaičiavimą įkeliant vaizdus, taip sumažinant įkėlimui reikiamą laiką. Jis remiasi rankiniu vardų užpildymu QGIS-

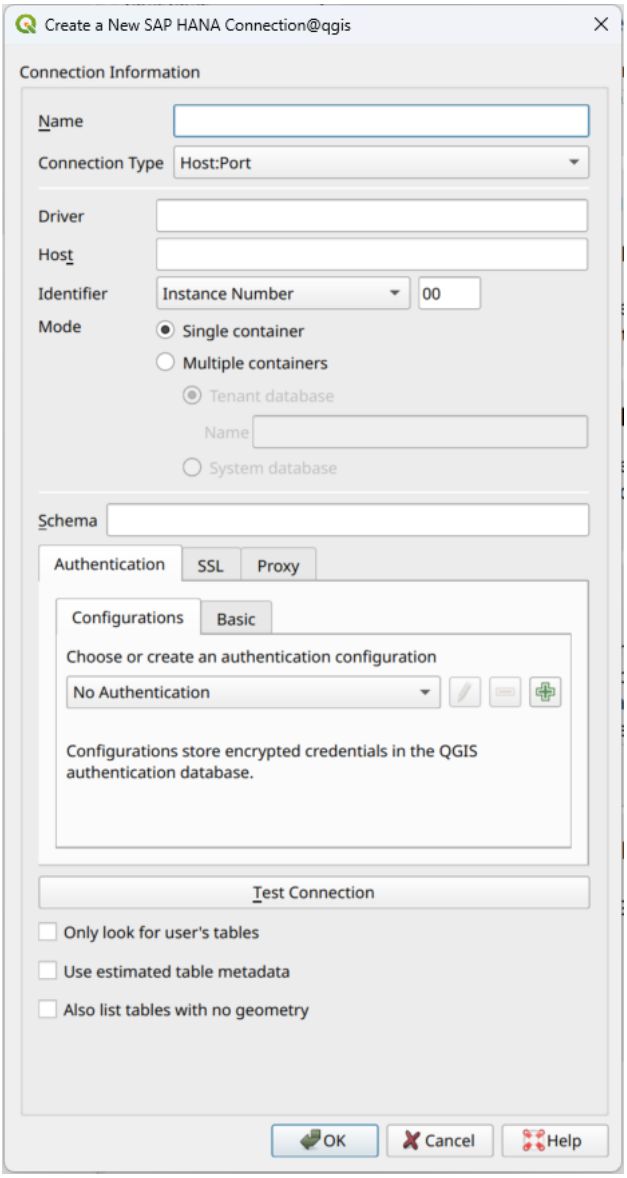
specifiniame `qgis_pkey` stulpelyje lentelėje `geometry_columns`. Jei pirminiam raktui naudojamas daugiau nei vienas stulpelis, jie turėtų būti užpildyti kaip kableliais atskirtos reikšmės.

- *Taipogi rodyti lenteles be geometrijos*: lentelės be prisegto geometrijos lauko irgi bus rodomos galimų lentelių sąraše.
- *Naudoti įvertintus lentelės parametrus*: bus naudojami tik įvertinti lentelės metaduomenys. Taip išvengiama lėtų lentelės skenavimų, bet dėl to gali būti gautos netikslios sluoksnio savybės, tokios kaip sluoksnio apimtis.
- *Praleisti netinkamų geometrijų apdorojimą*: visas įrašų su netinkamomis geometrijomis apdorojimas bus išjungtas. Tai pagreitina tiekėją, bet bet kokios lentelėje esančios netinkamos geometrijos reikš neprognozuojamus ir kartais trūkstamus įrašus. Įjunkite šią parinktį tik kai esate įsitikinę, kad visos duomenų bazėje esančios geometrijos yra tinkamos ir bet kokios naujai pridėtos geometrijos ar lentelės taipogi bus tinkamos.
- *Naudoti tik schemų poaibį* leis jums filtruoti MS SQL jungties schemas. Įjungus šią parinktį bus rodomos tik pažymėtos schemos. Jūs galite spausti *Įjungti* ar *Išjungti* bet kurią sąraše esančią schemą.

Prisijungimas prie SAP HANA

Pastaba

Jei norite prisijungti prie SAP HANA duomenų bazės, jums reikia SAP HANA kliento. Atsisiųsti SAP HANA klientą jūsų platformai galite iš SAP vystymo įrankių svetainės.



11.15 Fig. : Naujos SAP HANA jungties kūrimo dialogas

Galima įvesti šiuos parametrus:

- *Pavadinimas*: šios jungties pavadinimas.
- *Tvarkyklė* : HANA ODBC tvarkyklės pavadinimas. Jis yra `HDBODBC`, jei naudojate 64-bitų QGIS, `HDBODBC32`, jei naudojate 32-bitų QGIS. Atitinkamas tvarkyklės pavadinimas įvedamas automatiškai.
- *Tvarkyklė* : arba pavadinimas, kuriuo buvo registruota SAP HANA ODBC tvarkyklė `/etc/odbcinst.ini`, arba pilnas kelias iki SAP HANA ODBC tvarkyklės. Pagal nutylėjimą SAP HANA kliento diegyklė įdiegs ODBC tvarkyklę į `/usr/sap/hdbclient/libodbcHDB.so`.
- *Serveris*: Duomenų bazės serverio pavadinimas.
- *Identifikatorius*: identifikuoja serverio egzempliorių, prie kurio reikia prisijungti. Tai gali būti *Egzemplioriaus numeris* ar *Prievado numeris*. Egzemplioriaus numerį sudaro du skaitmenys, prievadai yra intervale nuo 1 iki 65,535.
- *Režimas*: nurodo kokiu režimu veikia SAP HANA egzempliorius. Šis nustatymas naudojamas tik tada, jei paskyros *Identifikatorius* nurodytas *Egzemplioriaus numeris*. Jei duomenų bazėje yra keli konteineriai, jūs galite arba prisijungti prie nuomininko su vardu, nurodytu *Nuomininko duomenų bazė*, arba galite prisijungti prie sisteminės duomenų bazės.

- *Schema*: šis parametras nėra privalomas. Jei nurodyta schema, QGIS duomenų ieškos tik toje schemoje. Jei šis laukas bus paliktas tuščias, QGIS duomenų ieškos visose schemose.
- *Authentication*: For general details about the authentication dialog behavior, see *Autentifikacija*.
- *SSL nustatymai*
  - *Įjungti TLS/SSL šifravimą*: įjungia TLS 1.1 - TLS1.2 šifravimą. Serveris pasirinks aukščiausią galimą.
  - *Tiekėjas*: nurodo kriptografinės bibliotekos tiekėją, kurį naudoja SSL komunikacija. *sapcrypto* turėtų veikti visose platformose, *openssl* turėtų veikti , *mscrypto* turėtų veikti ir *commoncrypto* reikalauja, kad būtų įdiegtas CommonCryptoLib.
  - *Tikrinti SSL sertifikatą*: įjungus šią parinktį, SSL sertifikatas bus tikrinamas naudojant pasitikėjimo saugyklą, kuri nurodyta *Pasitikėjimo saugojimo faile su viešu raktu*.
  - *Permušti mašinos pavadinimą sertifikate*: nurodo mašinos pavadinimą, kurį reikia naudoti tikrinant serverio identitetą. Čia nurodytas mašinos vardas tikrina serverio identitetą, o ne mašinos pavadinimą, kuriuo jungtis buvo sukurta. Jei jūs kaip mašinos pavadinimą nurodysite `*`, tada serverio mašinos pavadinimas nebus tikrinamas. Kiti pakeitimo ženklai neleidžiami.
  - *Raktų saugyklos failas su privačiu raktu*: šiuo metu ignoruojama. Šis parametras galėtų ateityje leisti autentifikuotis naudojant sertifikatą vietoje naudotojo vardo ir slaptažodžio.
  - *Pasitikėjimo saugyklos failas su viešu raktu*: nurodo kelią iki pasitikėjimo saugyklos failo, kuriame yra serverio vieši sertifikatai, jei naudojama OpenSSL. Paprastai pasitikėjimo saugykloje yra šakninis sertifikatas arba sertifikatas įstaigos, kuri pasirašė serverio viešus sertifikatus. Jei jūs naudojate kriptografijos biblioteką CommonCryptoLib ar msCrypto, tada palikite šią parinktį tuščią.
- *Žiūrėti tik naudotojo lentelėse*: įjungus šią parinktį QGIS ieškos tik lentelių ir vaizdų, kurių savininkas yra naudotojas, kuris jungiasi prie duomenų bazės.
- *Naudoti įvertintus lentelės metaduomenis*: įjungus šią parinktį, bus naudojami įvertinti lentelės metaduomenys, jei jie yra surinkti. Didelėms lentelėms taip išvengiama lėto lentelės įkėlimo ir potencialiai brangių skaičiavimų, bet gali reikšti neteisingas sluoksnio savybes, tokias kaip sluoksnio apimtis. Greitas apimties įvertinimas galimas pradedant QRC1/2024 ir SP8 su HANA Cloud ir atitinkamai HANA vietiniame.
- *Taipogi rodyti lenteles be geometrijų*: įjungus šią parinktį QGIS ieško ir lentelių bei vaizdų, kuriuose nėra erdvinių stulpelių.

Patarimas

Prisijungimas prie SAP HANA debesies

Jei norite prisijungti prie SAP HANA debesies egzemplioriaus, jums paprastai reikia nurodyti *Prievado numerį* `443` ir įjungti parinktį *Įjungti TLS/SSL šifravimą*.

Duomenų bazės sluoksnio įkėlimas

Once you have one or more connections defined to a database (see section *Įrašytos jungties kūrimas*), you can load layers from it. Of course, this requires that data are available. See section *Importing Data into PostgreSQL* for a discussion on importing data into a PostgreSQL database.

Kad įkeltumėte sluoksnį iš duomenų bazės, jums reikia įvykdyti šiuos žingsnius:

1. Atidarykite atitinkamos duomenų bazės kortelę dialoge *Duomenų šaltinių tvarkyklė*.
2. Iškrentančiame sąraše parinkite jungties pavadinimą ir spauskite *Prisijungti*.
3. The table below will be filled with your data grouped by schema, with a number of metadata information helpful for loading.

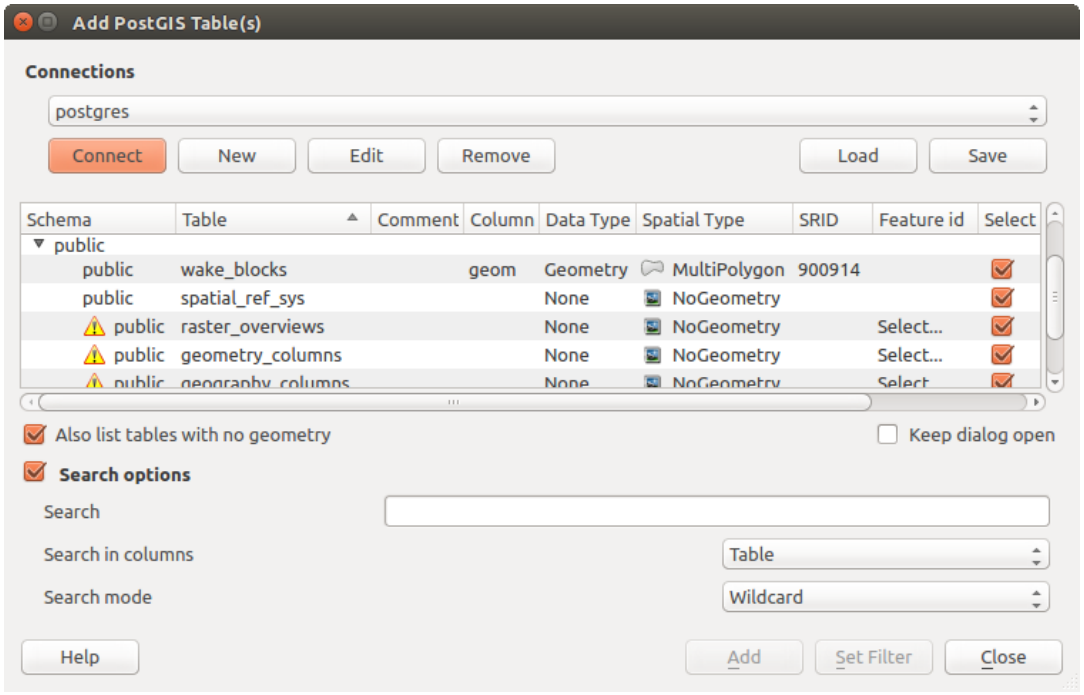
4. Įjunkite ar įjunkite parinktį *Taipogi rodyti lenteles be geometrijos*.
5. Pasirinktinai naudokite kelias *Paieškos parinktis*, kad sutrumpintumėte lentelių sąrašą iki to, kuris atitinka jūsų kriterijus. Jūs taipogi galite nurodyti šią parinktį prieš paspaudžiant mygtuką *Prisijungti*, kad pagreitintumėte informacijos ištraukimą iš duomenų bazės.
6. Prieinamų sluoksnių sąraše suraskite sluoksnį(ius), kurį norite pridėti.
7. Parinkite jį paspaudimu. Jūs galite parinkti kelis sluoksnius, parinkdami laikydami mygtuką `Shift` ar `Ctrl`.

   Layers can be selected only if they have no warning icon at the left-hand side of their first column. This may indicate an issue to detect:

   - features geometry type: in which case you can select the appropriate one in the drop-dowmn list of the *Spatial Type* column
   - layer CRS: you can enter the correct code in the *SRID* column
   - layer's primary key, in order to unequivocally identify each feature: this can be fixed by selecting one or more attributes in the drop-down list at the corresponding *Feature id* column.

   > **Patarimas**
   >
   > **Use the first column to store „primary keys" for views**
   >
   > Since PostgreSQL views don't support primary keys, a unique attribute or combination of attributes should always be selected. To help users to speed-up workflows, QGIS automatically selects the first attribute in the view. Therefore, users can define their views in a way that a unique column is in the first position of the view's definition. In this way, the view will be loaded with no extra interaction and the warning icon will never appear.

8. Jei taikoma, naudokite mygtuką *Nustatyti filtrą* (ar du kartus spauskite sluoksnį), kad įjungtumėte dialogą *Užklausos kūrėjas* (žr. skiltį *Query Builder*) ir nurodykite, kuriuos geoobjektus reikia įkelti iš parinkto sluoksnio. Filtro išraiška bus rodoma stulpelyje `sql`. Šį apribojimą galima išimti ar keisti *Sluoksnio savybės* ► *Bendra* ► *Tiekėjo geoobjektų filtras*.
9. Varnelė stulpelyje `Parinkti pagal id`, kuri pagal nutylėjimą įjungta, ištraukia geoobjektų id be jų atributų, todėl dažniausiai pagreitina duomenų įkėlimą.
10. Spauskite mygtuką *Pridėti*, kad pridėtumėte sluoksnį į žemėlapį.



11.16 Fig. : Add PostgreSQL Table(s) Dialog

Patarimas

Naudokite naršyklės skydelį, kad pagreitintumėte duomenų bazės lentelės(ių) įkėlimą

DB lentelių pridėjimas naudojant *Duomenų šaltinių tvarkyklę* kartais gali užimti daug laiko, nes QGIS kiekvienai lentelei iš karto ištraukia statistikas ir savybes (pvz. geometrijos tipą ir lauką, CRS, geoobjektų skaičių). Kad to išvengtumėte, kai *jungtis nustatyta*, geriau naudoti *Naršyklės skydelį* ar *DB tvarkyklę*, kad temptumėte ir numestumėte duomenų bazės lenteles į žemėlapio drobę.

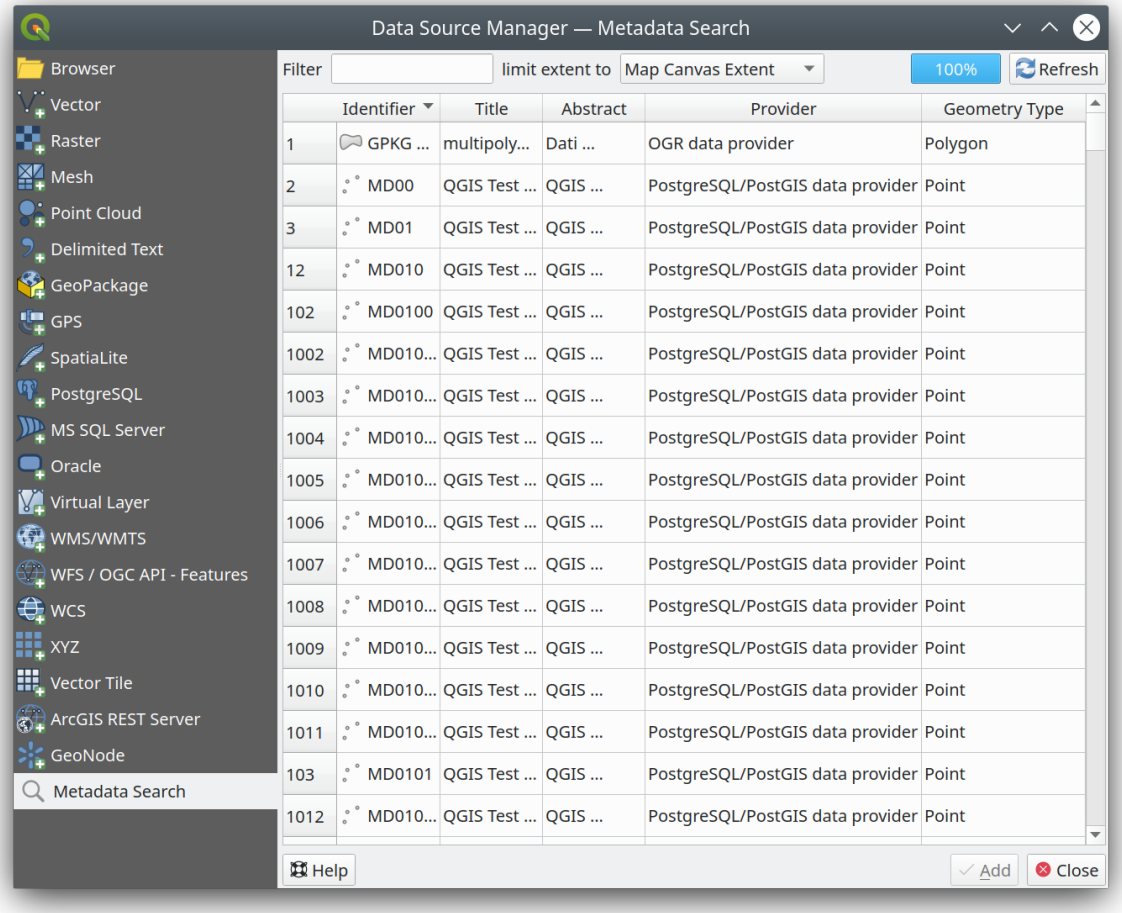
11.1.4 Sluoksnio metaduomenų paieškos skydelis

Pagal nutylėjimą QGIS gali gauti sluoksnio metaduomenis iš jungties ar duomenų tiekėjų, kurie leidžia saugoti metaduomenis (daugiau informacijos rasite *metaduomenų įrašymas į duomenų bazę*). Skydelis *Metaduomenų paieška* leidžia naršyti po sluoksnius pagal jų metaduomenis ir pridėti juos į projektą (arba du kartus paspaudus arba naudojant mygtuką *Pridėti*). Sąrašą galima filtruoti:

- pagal tekstą, stebint metaduomenų savybių aibę (identifikatorių, pavadinimą, santrauką)
- pagal erdvinę apimtį, naudojant dabartinio *projekto apimtį* arba žemėlapio drobės apimtį
- pagal sluoksnio (geometrijos) tipą

Pastaba

Metaduomenų šaltinis įgyvendinamas per sluoksnio metaduomenų tiekėjo sistemą, kurią gali išplėsti priedai.



11.17 Fig. : Sluoksnio metaduomenų paieškos skydelis

11.1.5 QGIS savi formatai

QGIS siūlo du savo formatus:

- Laikinas juodraštinis sluoksnis: atminties sluoksnis, kuris susietas su projektu (daugiau informacijos rasite skyriuje *Creating a new Temporary Scratch Layer*)
- Virtualūs sluoksniai: sluoksnis, kuris gaunamas įvykdžius kito sluoksnio(ių) užklausą (daugiau informacijos rasite skyriuje *Creating virtual layers*)

11.1.6 QLR - QGIS sluoksnio apibrėžimo failas

Sluoksnio apibrėžimus galima įrašyti kaip *Sluoksnio apibrėžimo failus* (QLR - `.qlr`) naudojant sluoksnio kontekstinį meniu *Eksportuoti ► Įrašyti kaip sluoksnio apibrėžimo failą…*.

QLR formatas leidžia dalintis „pilnais“ QGIS sluoksniais su kitais QGIS naudotojais. QLR failuose yra nuorodos į duomenų šaltinius ir visa QGIS stiliaus informacija, kurios reikia sluoksniui.

QLR failai rodomi naršyklės skydelyje ir juos galima naudoti, norint pridėti sluoksnius (su jų įrašytais stiliais) į sluoksnių skydelį. Jūs taipogi galite tempti ir numesti QLR failus iš sistemos failų tvarkyklės į žemėlapio drobę.

Galimi QLR failų veiksmai naršyklės skydelyje yra:

- *Eksportuoti sluoksnį ► Į failą*

- *Pridėti sluoksnį į projektą*
- *Sluoksnio savybės…*

11.1.7 Prisijungimas prie žiniatinklio paslaugų

QGIS leidžia jums pasiekti įvairius OGC žiniatinklio paslaugų tipus (WM(T)S, WFS(-T), WCS, CSW, …). QGIS Serverio dėka jūs galite taipogi ir patys publikuoti tokias paslaugas. QGIS-Server-manual rasite šių galimybių aprašymus.

Vektorinių kaladėlių paslaugų naudojimas

Vektorinių kaladėlių paslaugas galima pridėti kortelėje *Vektorinės kaladėlės*, kurią rasite dialoge *Duomenų šaltinių tvarkyklė* ar įrašo *Vektorinės kaladėlės* kontekstiniame meniu *Naršyklės* skydelyje. Paslaugos gali būti arba *Nauja bendra jungtis…* arba *Nauja ArcGIS vektorinių kaladėlių paslaugos jungtis…*.

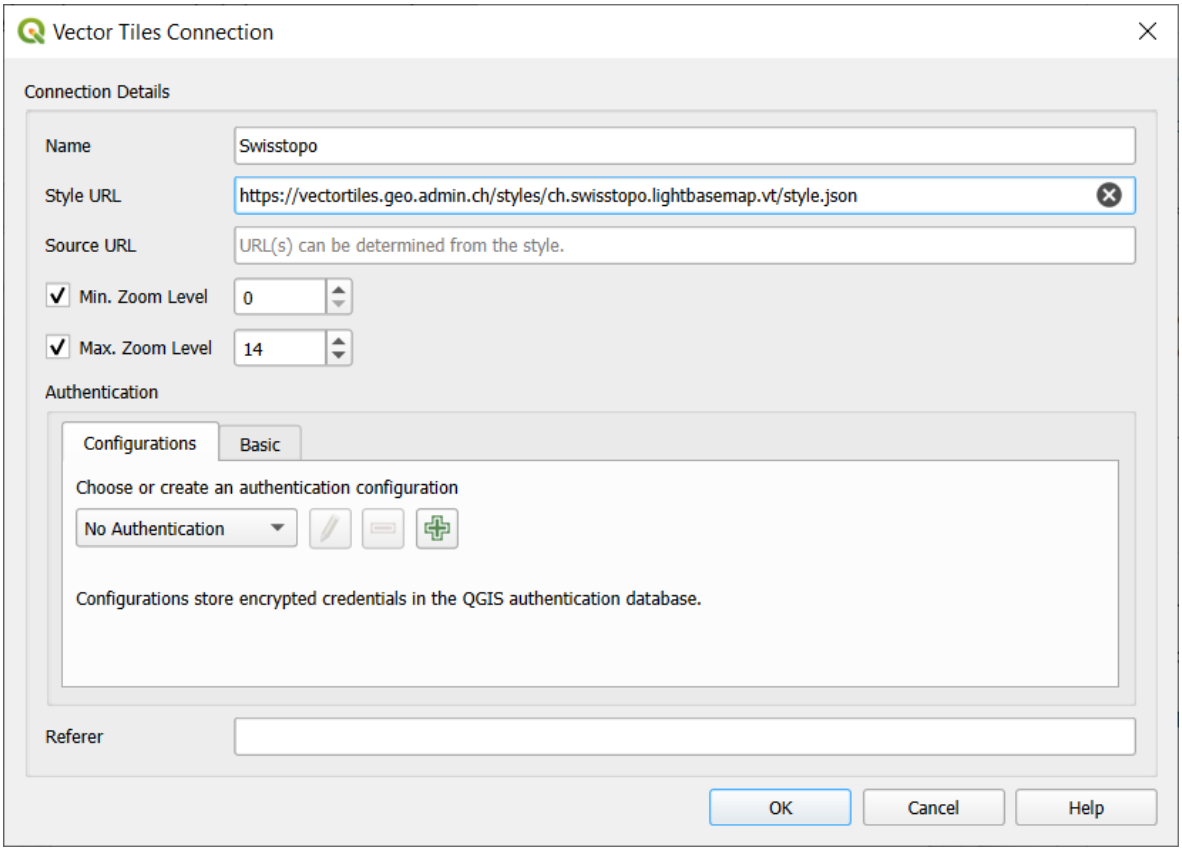
Jūs kuriate paslaugą pridėdami:

- *Pavadinimą*
- *Stiliaus URL*: URL į MapBox GL JSON stiliaus konfigūraciją. Jei pateikta, tai stilius bus taikomas, kai jungties sluoksniai bus pridėti į QGIS. Jei tai Arcgis vektorinių kaladėlių paslaugos jungtis, URL permuš numatytą stilių konfigūraciją, nurodytą serverio konfigūracijoje.

  Jūs galite įkelti vektorines kaladėles tiesiogiai iš *Stiliaus URL*. Duomenų šaltinis automatiškai perduodamas iš stiliaus, palaikomi URL su keliais šaltiniais. Dėl to *Šaltinio URL* yra neprivalomas.
- *Šaltinio URL*: `http://pavyzdys.lt/{z}/{x}/{y}.pbf` tipo bazinėms paslaugoms ir `http://pavyzdys.lt/arcgis/rest/services/Layer/VectorTileServer` ArcGIS paremtoms paslaugoms. Paslauga turi teikti kaladėles `.pbf` formatu.
- ☑ *Min. mastelis* ir ☑ *Maks. mastelis*. Vektorinės kaladėlės turi piramidės struktūrą. Naudodami šias parinktis jūs galite kurti individualius kaladėlių piramidės sluoksnius. Šis sluoksniai bus naudojami braižant vektorines kaladėles QGIS'e.

  Merkatoriaus projekcijai (kurią naudoja OpenStreetMap vektorinės kaladėlės) mastelio lygis 0 reprezentuoja visą pasaulį masteliu 1:500.000.000. Lygis 14 reprezentuoja mastelį 1:35.000.
- the *authentication* configuration if necessary
- *Nukreipėją*

Fig. 11.18 vaizduojamas dialogas su vektorinių kaladėlių paslaugos konfigūracija.



11.18 Fig. : Vektorinės kaladėlės - paslaugos konfigūravimas

Konfigūraciją galima įrašyti į `.XML` failą (*Įrašyti jungtis*) per įrašą *Vektorinės kaladėlės*, kurį rasite dialoge *Duomenų šaltinių tvarkyklė* arba per jo kontekstinį meniu skydelyje *Naršyklė*. Analogiškai ją galima ir pridėti iš failo (*Įkelti jungtis*).

Kai jungtis su vektorinių kaladėlių paslauga nustatyta, galima:

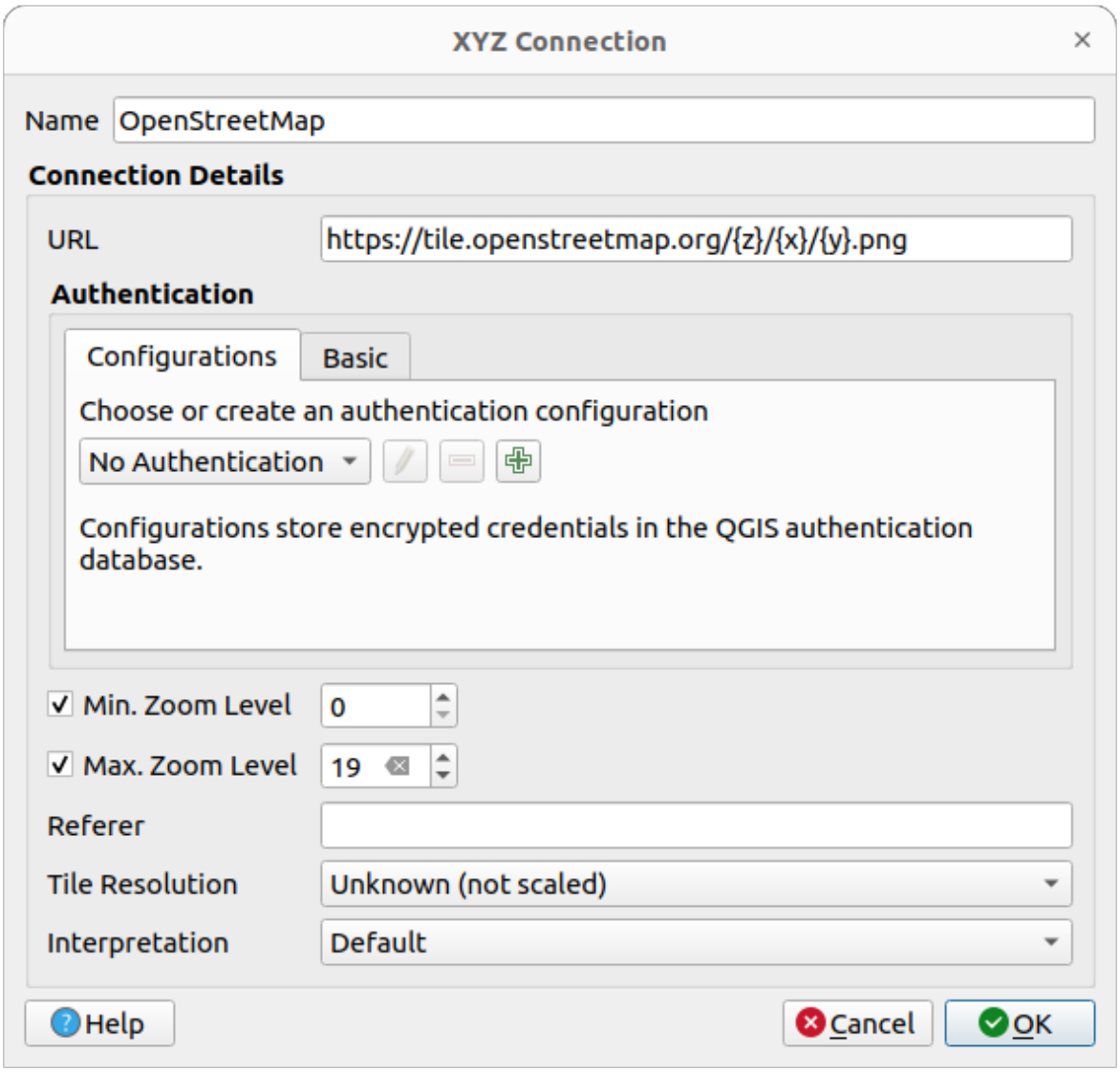
- *Keisti* vektorinių kaladėlių jungties nustatymus
- *Išimti* jungtį
- Skydelyje *Naršyklė* paspaudus dešiniu pelės mygtuku virš įrašo jūs taipogi galite:
  - *Pridėti sluoksnį į projektą*: dvigubas paspaudimas taipogi prideda sluoksnį
  - Žiūrėti *Sluoksnio savybes...* ir gauti prieigą prie metaduomenų bei peržiūrėti paslaugos teikiamus duomenis. Daugiau nustatymų rasite įkėlus sluoksnį į projektą.

XYZ kaladėlių paslaugų naudojimas

XYZ kaladėlių paslaugas galima pridėti kortelėje *XYZ*, kurią rasite dialoge *Duomenų šaltinių tvarkyklė* arba *XYZ kaladėlių* įrašo kontekstiniame meniu *Naršyklės* skydelyje. Pagal nutylėjimą QGIS teikia kelias naudojimui paruoštas XYZ kaladėlių paslaugas:

- *Mapzen Global Terrain*, iš karto suteikia projektams prieigą prie pasaulinio DEM šaltinio. Daugiau informacijos ir resursų rasite https://registry.opendata.aws/terrain-tiles/
- *OpenStreetMap* prieiga prie viso pasaulio 2D žemėlapio. Fig. 11.19 vaizduojamas dialogas su OpenStreetMap XYZ kaladėlių paslaugos konfigūracija.

Kad pridėtumėte naują paslaugą, spauskite *Nauja* (atitinkamai *Nauja jungtis* naršyklės skydelyje) ir pateikite:



11.19 Fig. : XYZ kaladėlės - OpenStreetMap konfigūracija

- *Pavadinimą*
- *URL*, jūs galite pridėti `http://pavyzdys.lt/{z}/{x}/{y}.png` ar `file:///vietinis_kelias/{z}/{x}/{y}.png`
- the *authentication* configuration if necessary
- *Min. mastelis* ir *Maks. mastelis*
- *Nukreipėją*
- *Kaladėlių rezoliucija*: galimos reikšmės yra *Nežinoma (nekeista)*, *Standartinė (256x256 / 96DPI)* ir *Aukšta (512x512 / 192DPI)*
- *Interpretavimas*: keičia WMTS/XYZ rastro duomenų aibes į vienos kintančio kablelio juostos rastro sluoksnį pagal iš anksto nustatytą kodavimo sistemą. Palaikomos schemos yra *Numatytoji* (nedaromas joks konvertavimas), *MapTiler paviršiaus RGB* ir *Terrarium paviršiaus RGB*. Parinktas keitimas išvers pradines RGB reikšmes į slankaus kablelio reikšmes kiekvienam pikseliui. Įkėlus, sluoksnis bus rodomas kaip vienos juostos slankaus kablelio rastras, paruoštas simbolizavimui naudojant standartinius QGIS *rastro braižymus*.

Spauskite *Gerai*, kad sukurtumėte jungtį. Tada bus galima:

- *Pridėti* į projektą naują sluoksnį, jis įkeliamas nustatymuose nurodytu pavadinimu.

- *Keisti* XYZ jungties nustatymus
- *Išimti* jungtį
- Skydelyje *Naršyklė* paspaudus dešiniu pelės mygtuku virš įrašo jūs taipogi galite:
  - *Eksportuoti sluoksnį…* ► *Į failą*, *įrašyti jį kaip rastrą*
  - *Pridėti sluoksnį į projektą*: dvigubas paspaudimas taipogi prideda sluoksnį
  - Žiūrėti *Sluoksnio savybes…* ir gauti prieigą prie metaduomenų bei peržiūrėti paslaugos teikiamus duomenis. Daugiau nustatymų rasite įkėlus sluoksnį į projektą.

Konfigūraciją galima įrašyti į `.XML` failą (*Įrašyti jungtis*) per įrašą *XYZ*, kurį rasite dialoge *Duomenų šaltinių tvarkyklę* arba per jo kontekstinį meniu skydelyje *Naršyklė*. Analogiškai ją galima ir pridėti iš failo (*Įkelti jungtis*).

OpenStreetMap skirtas XML failas atrodo taip:

```
<!DOCTYPE connections>
<qgsXYZTilesConnections version="1.0">
  <xyztiles url="https://tile.openstreetmap.org/{z}/{x}/{y}.png"
   zmin="0" zmax="19" tilePixelRatio="0" password="" name="OpenStreetMap"
   username="" authcfg="" referer=""/>
</qgsXYZTilesConnections>
```

Patarimas

XYZ kaladėlių įkėlimas nekuriant jungties

Galima pridėti į projektą XYZ kaladėles neįrašant jungties nustatymų į jūsų naudotojo profilį (pvz. duomenų rinkiniui, kurio jums reikia tik vieną kartą). Kortelėje *Duomenų šaltinių tvarkyklė* ► *XYZ*, keiskite bet kokias savybes grupėje *Jungties detalės*. Aukščiau esantis laukas *Pavadinimas* turėtų pasikeisti į `Savas`. Spauskite *Pridėti*, kad įkeltumėte sluoksnį į projektą. Pagal nutylėjimą jis bus pavadintas `XYZ sluoksnis`.

XYZ kaladėlių paslaugų pavyzdžiai:

- OpenStreetMap Monochrominis: *URL*: `http://tiles.wmflabs.org/bw-mapnik/{z}/{x}/{y}.png`, *Min. mastelis*: 0, *Maks. mastelis*: 19.
- Google Maps: *URL*: `https://mt1.google.com/vt/lyrs=m&x={x}&y={y}&z={z}`, *Min. mastelis*: 0, *Maks. mastelis*: 19.
- Open Weather Map temperatūra: *URL*: `http://tile.openweathermap.org/map/temp_new/{z}/{x}/{y}.png?appid={api_key}` *Min. mastelis*: 0, *Maks. mastelis*: 19.

ArcGIS REST serverių naudojimas

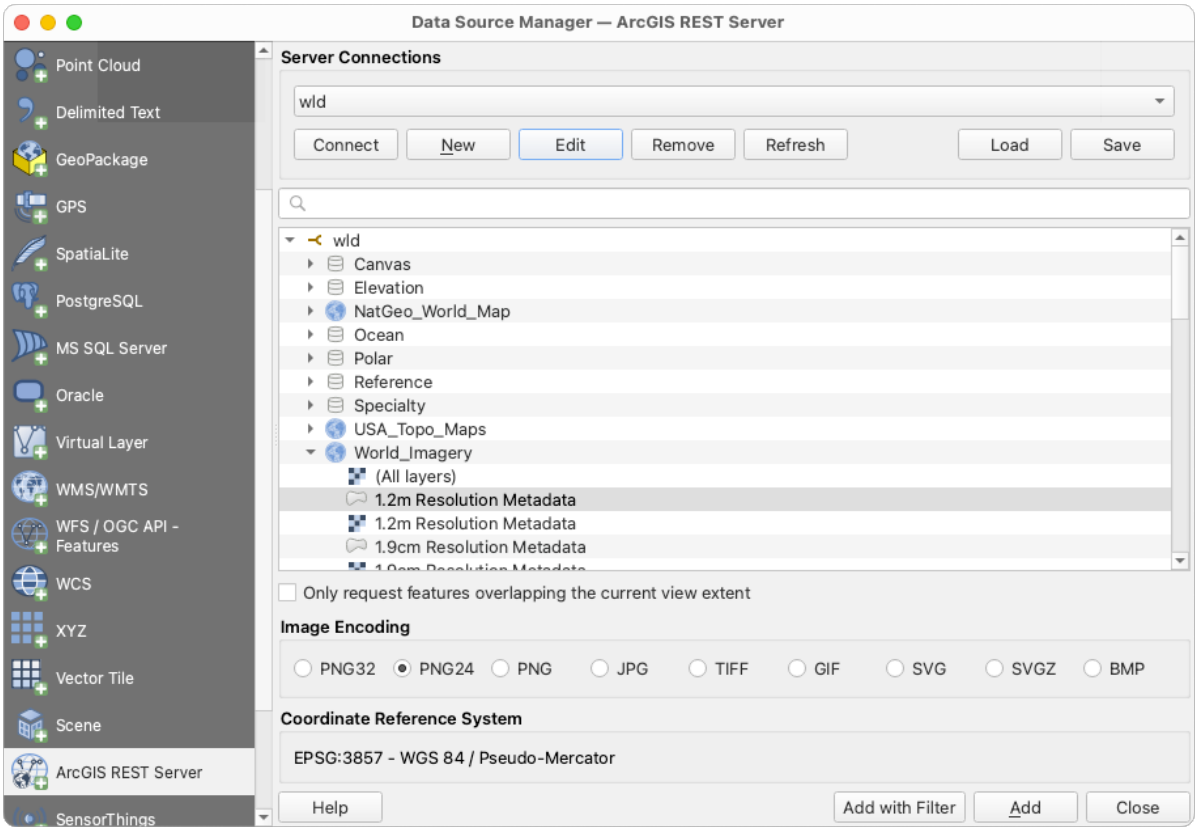
An ArcGIS REST Server can host many different types of web services (feature service, map service, image service, …). ArcGIS REST Servers can be added via the *ArcGIS REST Server* tab of the *Data Source Manager* dialog or the contextual menu of the *ArcGIS REST Servers* entry in the *Browser* panel:

1. Press *New* (respectively *New Connection*) and provide:
   - a *Name*: A name for the connection.
   - the *URL*: Main address of the ArcGIS REST Server.
   - a *Prefix*: Used to specify the proxy prefix in the URL, which is necessary for some ArcGIS servers that use web proxy prefixes.
   - a *Community endpoint URL*: Endpoint URL for ArcGIS Online or Portal for ArcGIS, used to access content groups.
   - a *Content endpoint URL*: Endpoint URL for the content service. This is used to access items in ArcGIS Online or Portal for ArcGIS.

- the *authentication* credentials if necessary.
- a *Referer*: The referer URL to be sent in the HTTP headers when making requests to the server. This may be required by some servers for authentication purposes.

> **Pastaba**
>
> ArcGIS Feature Service jungtys, kurios turi nustatytus savo atitinkamus Portalo prieigos taškų URL, gali būti tiriamos naršyklės skydelio turinio grupėse.
>
> Jei nurodyti jungties portalo prieigos taškai, tada išskleidus jungtį naršyklėje matysite „Grupių“ ir „Paslaugų“ aplankus vietoje paprastai rodomo pilno paslaugų sąrašo. Išskleidus grupių aplanką matysite sąrašą visų turinio grupių, kurių narys yra prisijungęs naudotojas, kiekvieną iš jų galima išskleisti ir pamatyti tai grupei priklausančius paslaugų elementus.

2. Then press *OK* to validate the configuration settings. These configurations can be saved to `.XML` file (*Save*). Likewise, they can be added from a file (*Load*).
3. Once a connection to an ArcGIS REST Server is set, it is possible to:
   - *Keisti* ArcGIS REST Serverio jungties nustatymus
   - *Išimti* jungtį
   - *Atnaujinti* jungtį
4. Press *Connect* to request the server and display its contents. They are organized in a tree structure whose nodes depend on the connection's endpoint.



11.20 Fig. : Data Source Manager - ArcGIS REST Server

5. Expand the tree to find and select the layers of interest. Their *Coordinate Reference System* is displayed at the bottom of the dialog.

6. For raster-based layers, you can select the *Image encoding* to use among a number of image formats advertized by the target service (e.g., `PNG`, `JPG`, `GIF`, `SVG`, `SVGZ`, …).
7. To add the selected layers to the map canvas, press *Add* button.

   Because layers can sometimes load and render slowly on the client side, applying a filter to restrict the features retrieved from the service can significantly improve performance, since only the filtered features are requested from the server. This can be done by:

   - checking *Only request features overlapping the current view extent*
   - for vector layers (feature service), pressing *Add with filter* to apply attribute-based filters to the layer with *Išraiškų teksto kūrėjas* functions.

     In the *Browser* panel, right-click the ArcGIS REST Server layer and select *Add Filtered Layer to Project* will also open the builder dialog.

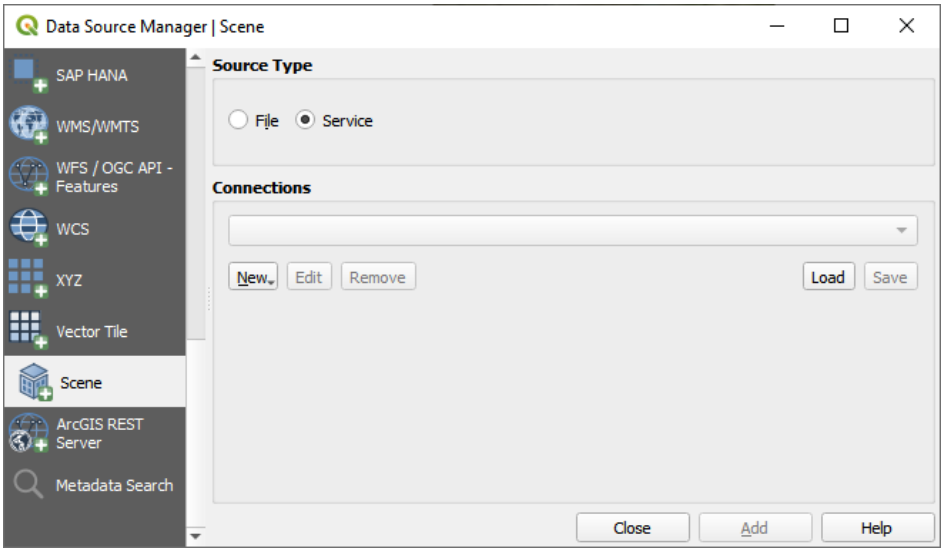
Most of the above tools to connect and access layers on an ArcGIS REST Server are also available from the *Browser* panel, within the contextual menu of the target node item. Depending on the node, you can also:

- *View Service Info* which will open the default web browser and display information on the requested service.
- View the *Layer Properties…* and get access to the metadata and a preview of the data provided by the service. More settings are available when the layer has been loaded into the project.

3D scenų kaladėlių paslaugų naudojimas

QGIS palaiko įvairius 3D scenų kaladėlių formatus, sugrupuotus kaip „scenų kaladėlės“. Tarp jų yra 3D kaladėlės ir Quantized Mesh kaladėlės.

Norėdami įkelti scenos kaladėlių duomenų aibę į QGIS, naudokite kortelę *Scena*, kurią rasite dialoge *Duomenų šaltinių tvarkyklė*.



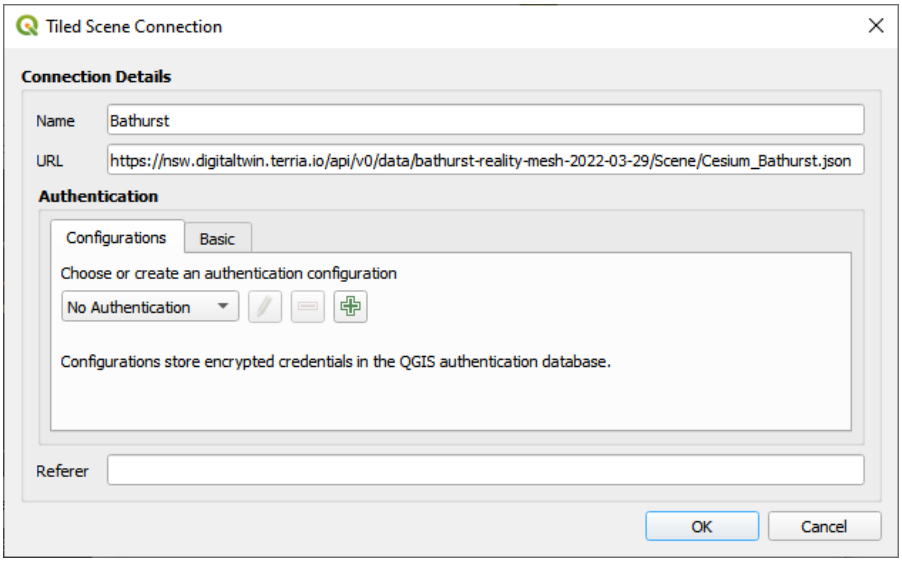
11.21 Fig. : Duomenų šaltinių tvarkyklė - scena

Sukurkite jungtį spausdami mygtuką *Nauja*. Jūs galite pridėti *Naują Cesium 3D kaladėlių jungtį* arba *Naują Quantized Mesh jungtį*.

Parinkite *Vardą* ir nustatykite *URL* tokį, kuris atitinka sluoksnio aprašymą JSON faile.

URL gali būti nutolęs (pvz. `http://example.com/tileset.json`) arba vietinis (t.y. `file:///path/to/tiles/tileset.json`).

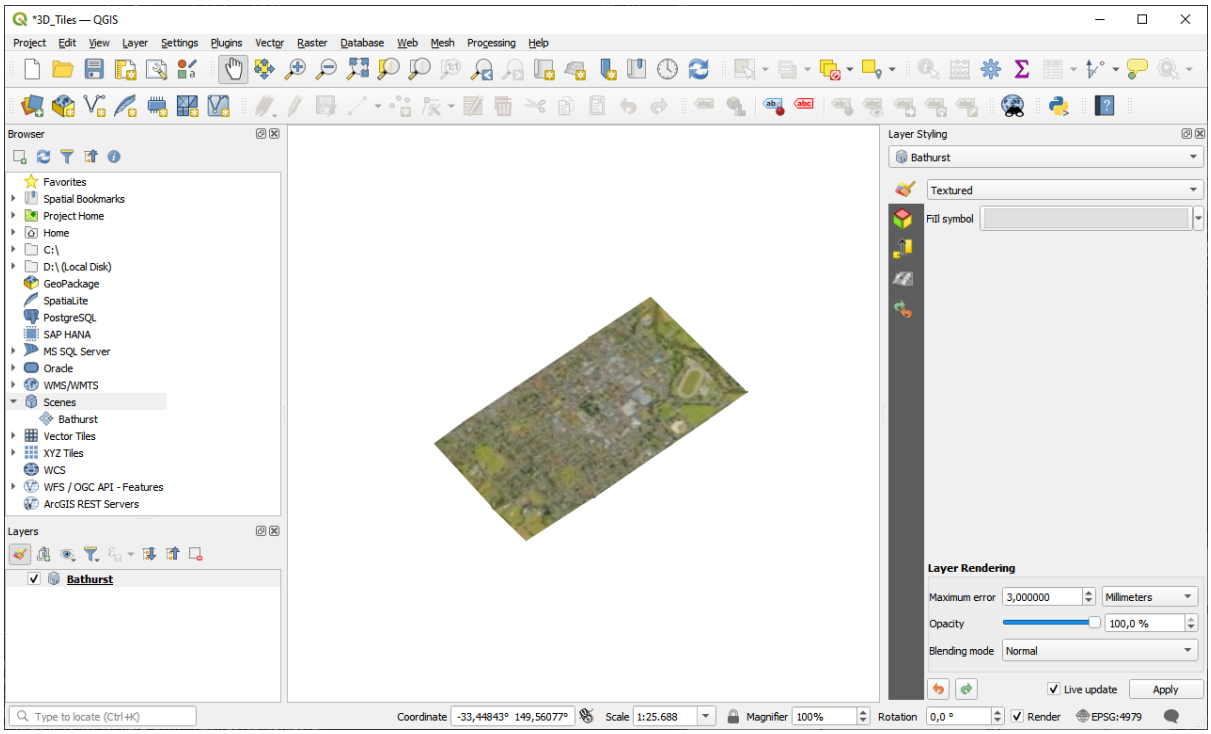
The *authentication* configuration if necessary.



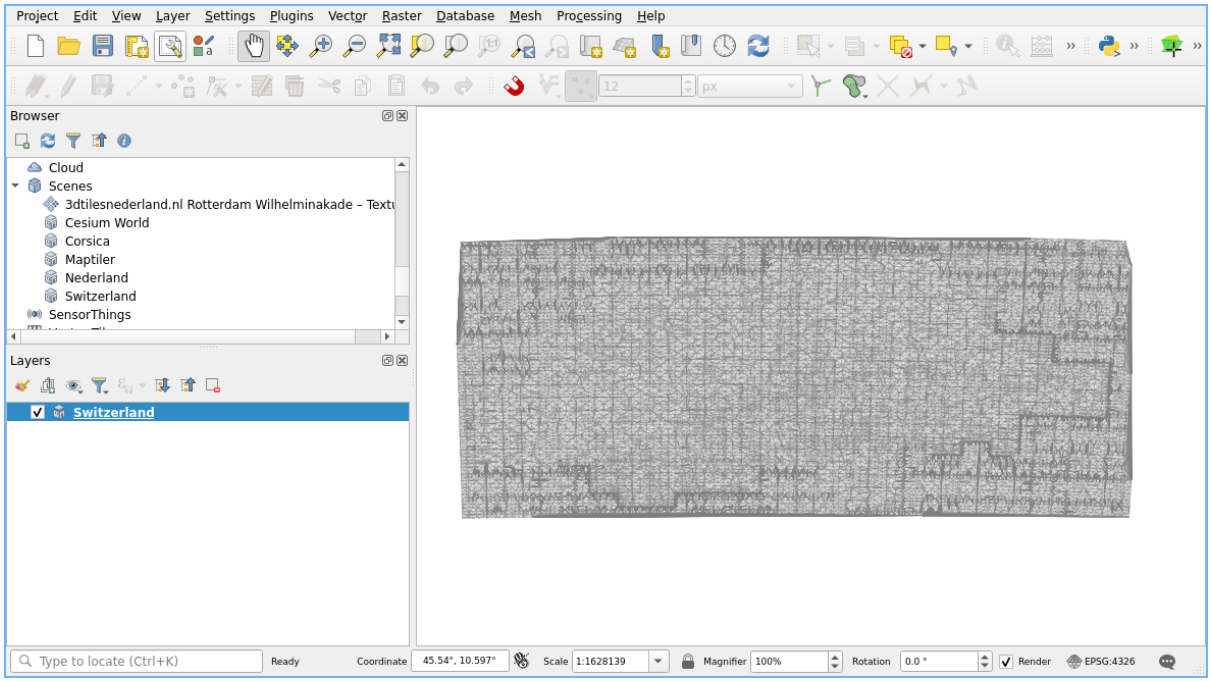
11.22 Fig. : Scenos kaladėlių jungtis

Jūs taipogi galite pridėti paslaugą per *Naršyklės skydelį*.

Sukūrę naują jungtį galite *Pridėti* naują sluoksnį į jūsų žemėlapį.



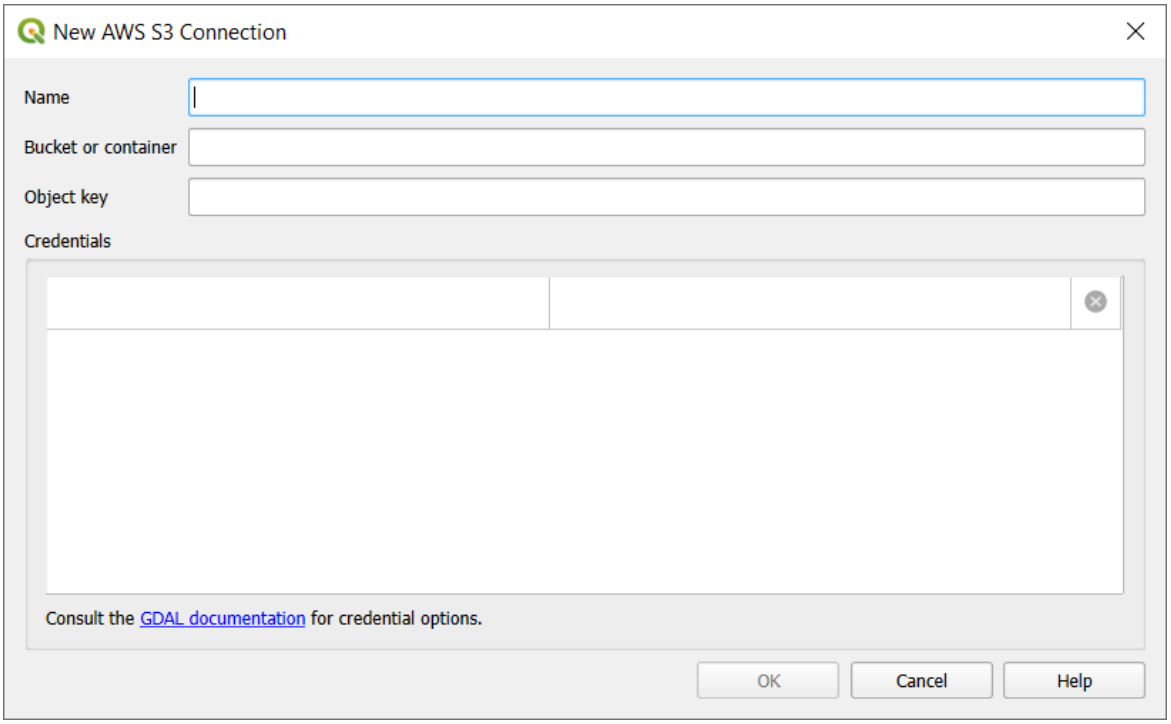
11.23 Fig. : 3D kaladėlių sluoksnis - su tekstūromis



11.24 Fig. : Kvantizuotas tinklo sluoksnis

Debesies jungčių naudojimas

QGIS palaiko jungtis į debesijos paslaugas, tokias kaip Alibaba Cloud OSS, Amazon S3, Google Cloud Storage, Microsoft Azure Blob Storage, Microsoft Azure Data Lake Storage ir OpenStack Swift Object Storage. Jūs galite įkelti vektorinius ir rastro duomenis iš šių paslaugų į QGIS. Sukurkite naują *Debesijos* jungtį *Naršyklės* skydelyje spausdami dešinį mygtuką ant įrašo *Debesija* ir parinkdami *Nauja jungtis*. Jūs pamatysite galimų debesijos paslaugų iškrentantį sąrašą. Parinkite jums reikiamą paslaugą, prie kurios norite prisijungti, ir užpildykite reikiamus laukus:



11.25 Fig. : Debesijos prisijungimo dialogas

- *Pavadinimas*: Jungties pavadinimas.
- *Kibiras arba konteineris*: debesijos paslaugų kibiro ar konteinerio pavadinimas.
- *Objekto raktas* (neprivalomas): raktas objekto kibire ar konteineryje.
- *Prisijungimo duomenys*: prieigos prie debesijos paslaugos prisijungimo duomenys.

Jūs taipogi galite pasirinkti *Įrašyti jungtį* į XML failą *Įkelti jungtį* iš XML failo.

11.2 Sluoksnių kūrimas

Sluoksnius galima kurti įvairiais būdais, pavyzdžiui:

- kurti tuščią sluoksnį nuo nulio
- kurti sluoksnį pagal esamus sluoksnius
- kurti sluoksnį iš iškarpinės
- sluoksniai kaip SQL-tipo užklausos rezultatas iš vieno ar daugiau sluoksnių (*virtualūs sluoksniai*)

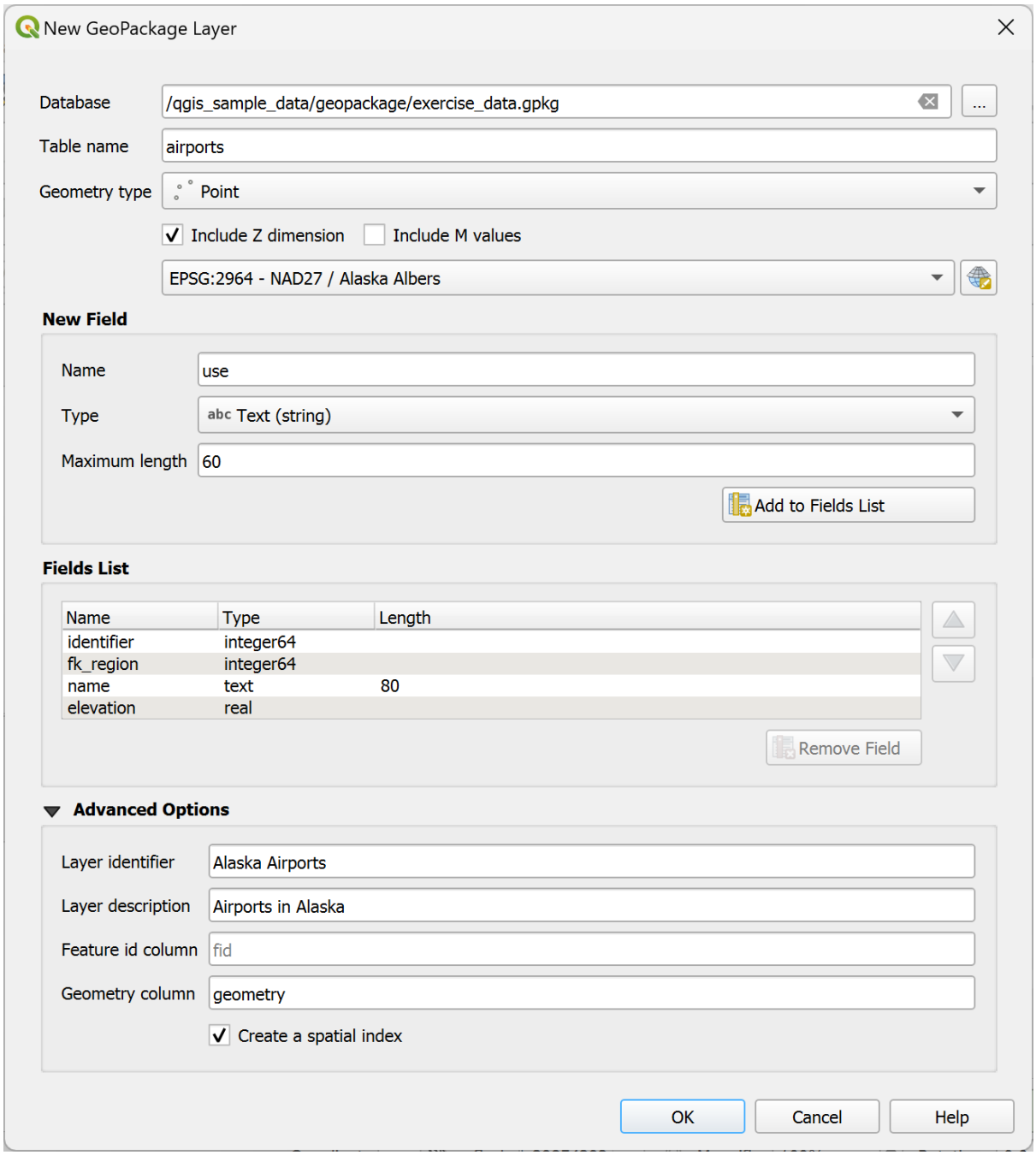
QGIS taipogi teikia įrankius importavimui/eksportavimui iš/į skirtingus formatus.

11.2.1 Naujų vektorinių sluoksnių kūrimas

QGIS leidžia jums kurti naujus sluoksnius skirtingais formatais. Jis teikia įrankius, leidžiančius kurti GeoPackage, Shapefile, SpatiaLite, GPX formatais ir Laikinus juodraštinius sluoksnius (dar žinomus kaip atminties sluoksniai). *Naujo GRASS sluoksnio* kūrimas palaikomas iš GRASS priedo.

Naujo GeoPackage sluoksnio kūrimas

Kad sukurtumėte naują GeoPackage sluoksnį, spauskite mygtuką *Naujas GeoPackage sluoksnis…*, kurį rasite meniu *Sluoksnis* ► *Kurti sluoksnį* ► arba įrankinėje *Duomenų šaltinių tvarkyklė*. Taipogi galite sukurti naują GeoPackage sluoksnį *Naršyklės skydelyje* parinktę *Kurti duomenų bazę ir sluoksnį…*. Bus parodytas dialogas *Naujas GeoPackage sluoksnis*, kaip matome paveiksle Fig. 11.26.



11.26 Fig. : Naujo GeoPackage sluoksnio kūrimo dialogas

1. Pirmas žingsnis yra nurodyti duomenų bazės failo vietą. Tai galima padaryti paspaudus mygtuką …, kuris yra lauko *Duombazė* dešinėje ir parinkus esamą GeoPackage failą arba sukūrus naują. QGIS automatiškai pridės reikiamą plėtinį prie jūsų nurodyto pavadinimo.
2. Nurodykite naujo sluoksnio / lentelės pavadinimą (*Lentelės pavadinimas*)
3. Apibrėžkite *Geometrijos tipą*. Jei tai nėra sluoksnis be geometrijos, jūs galite nurodyti, ar jame turi reikia *Įtraukti Z matmenį* ir/ar *Įtraukti M reikšmes*.
4. Naudodami mygtuką nurodykite koordinačių atskaitos sistemą

Norėdami pridėti laukus į jūsų kuriamą sluoksnį:

1. Įveskite lauko *Pavadinimą*
2. Parinkite duomenų *Tipą*. Palaikomi tipai yra *Tekstiniai duomenys*, *Sveiki skaičiai* (integer ir integer64), *Dešimtainiai skaičiai*, *Datos* ir *Datos su lauku*, *Dvejetainiai (BLOB)* ir *Loginiai*.
3. Priklausomai nuo parinkto duomenų formato, įveskite reikšmių *Maksimalų ilgį*.
4. Spauskite mygtuką *Pridėti į laukų sąrašą*
5. Kartokite aukščiau pateiktus veiksmus visiems pridedamiems laukams
6. Vėliau galite keisti laukų rikiuotę naudodami mygtukus Perkelti aukštyn ir Perkelti žemyn
7. Kai jums jau tinka atributai, spauskite *Gerai*. QGIS pridės naują sluoksnį į legendą ir jūs galėsite redaguoti jį kaip aprašyta skyriuje *Digitizing an existing layer*.

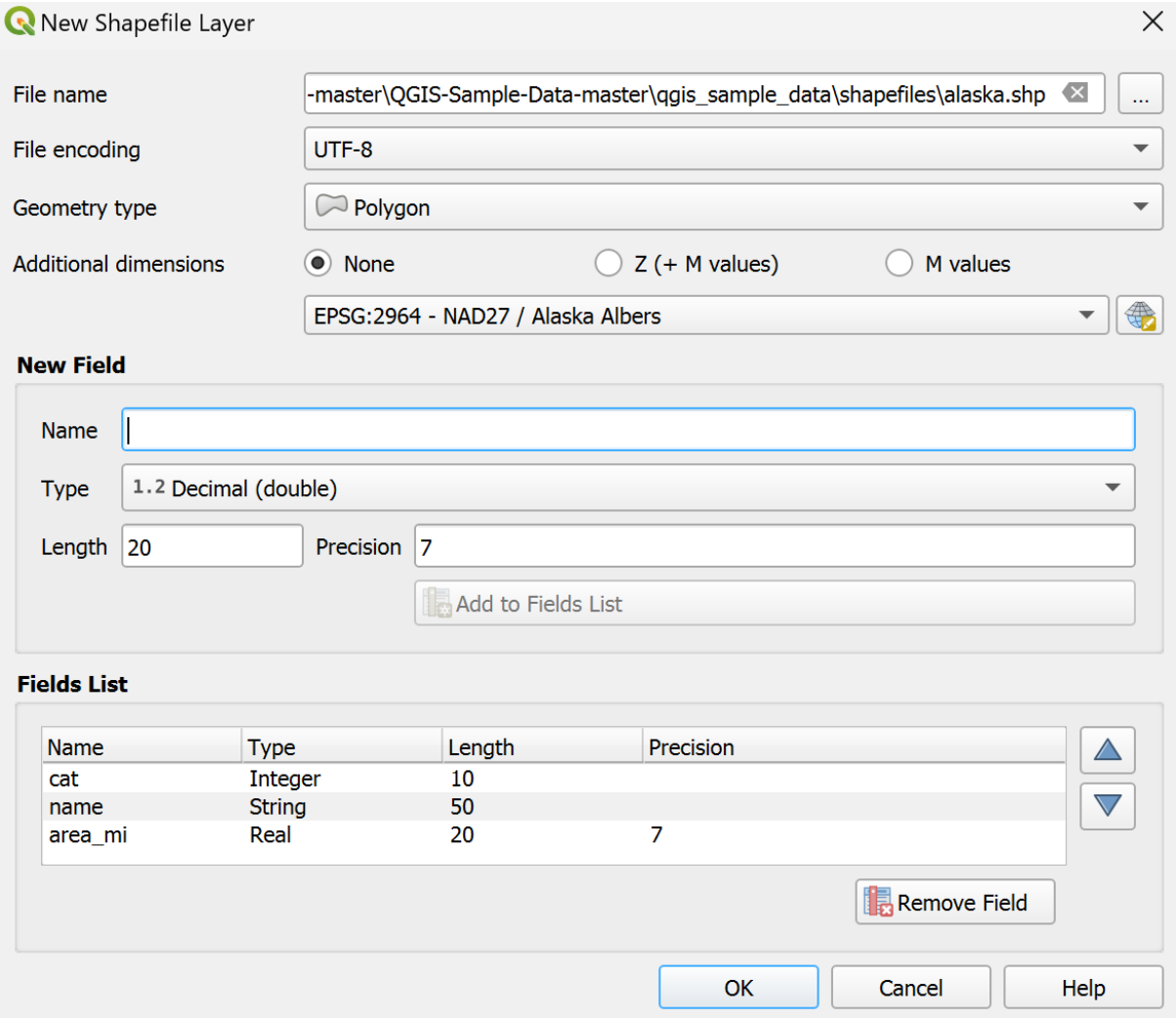
Pagal nutylėjimą, kuriant GeoPackage sluoksnį QGIS sukuria *Geoobjekto id lauką* pavadinimu `fid`, kuris veikia kaip sluoksnio pirminis raktas. Pavadinimą galima pakeisti. Geometrijos laukas, jei toks yra, vadinamas `geometry` ir jūs galite pasirinkti *Sukurti jam erdvinį indeksą*. Šias parinktis rasite tarp *Sudėtingesnių parinkčių*, kartu su *Sluoksnio identifikatoriumi* (trumpu žmogui skaitomu sluoksnio pavadinimu) bei *Sluoksnio aprašymu*.

Tolimesnį GeoPackage sluoksnių tvarkymą galima atlikti su *DB tvarkykle*.

Naujo Shape failų sluoksnio kūrimas

Norėdami sukurti naują ESRI Shape failų formato sluoksnį, spauskite mygtuką *Naujas Shape failų sluoksnis…* meniu *Sluoksnis* ► *Kurti sluoksnį* ► arba per *Duomenų šaltinių tvarkyklės* įrankinėje. Dialogas *Naujas Shape failo sluoksnis* bus rodomas kaip parodyta Fig. 11.27.

1. Nurodykite kelią ir failo pavadinimą naudodami mygtuką …, esantį greta *Failo pavadinimas*. QGIS automatiškai pridės tinkamą praplėtimą prie jūsų pateikto pavadinimo.
2. Tada nurodykite duomenų *Failo koduotę*
3. Parinkite sluoksnio *Geometrijos tipą*: Be geometrijos (tai reikš `.DBF` formato failą), taškas, multitaškas, linija arba poligonas
4. Nurodykite ar geometrija turi turėti papildomus matmenis: *Jokių*, *Z (+ M reikšmės)* ar *M reikšmės*
5. Naudodami mygtuką nurodykite koordinačių atskaitos sistemą



11.27 Fig. : Shape failo sluoksnio kūrimo dialogas

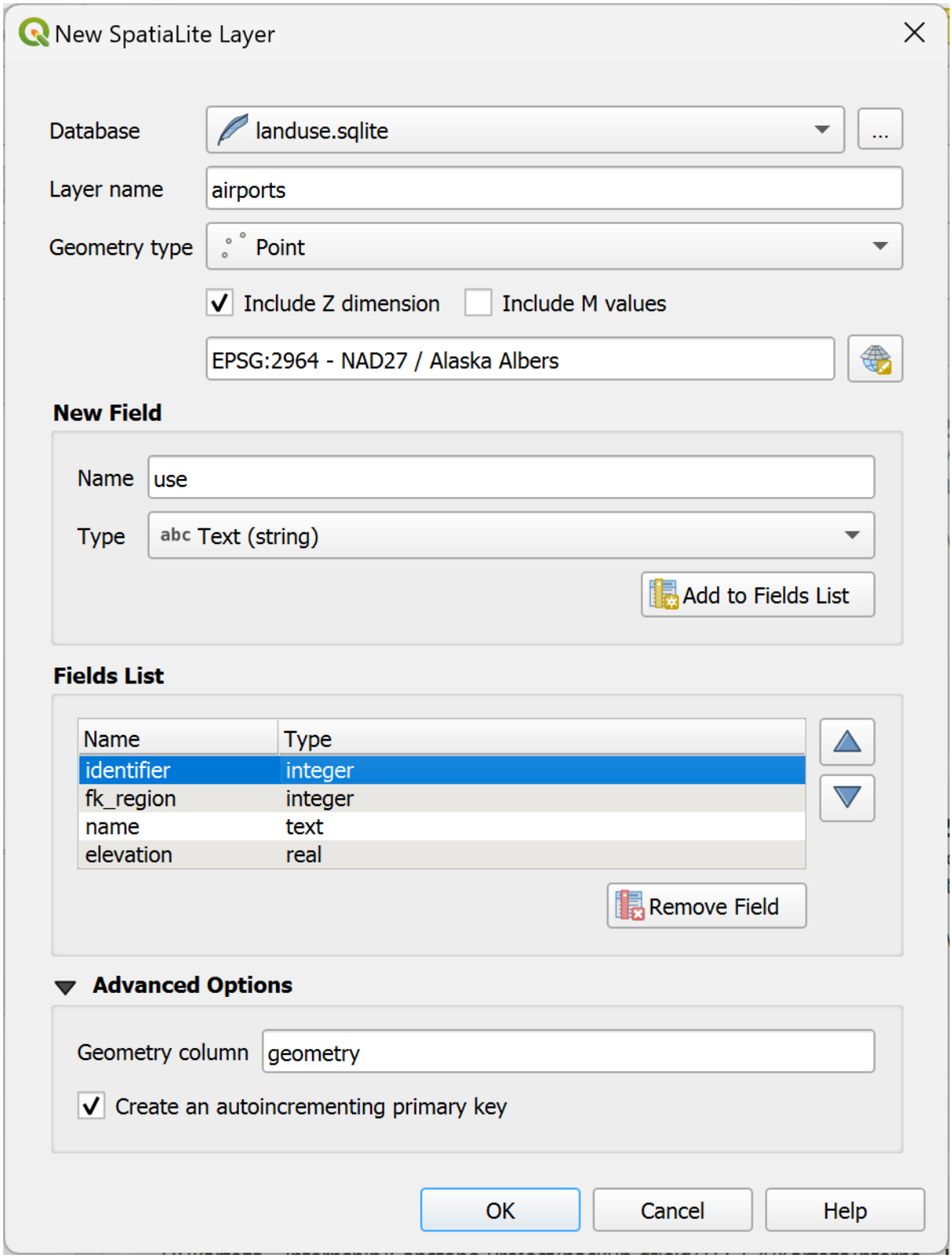
Norėdami pridėti laukus į jūsų kuriamą sluoksnį:

1. Įveskite lauko *Pavadinimą*
2. Parinkite duomenų *Tipą*. Palaikomi tik *Dešimtainių skaičių*, *Sveikų skaičių*, *Tekstinių duomenų*, *Datos* ir *Loginiai* duomenų tipų atributai.
3. Priklausomai nuo parinkto duomenų formato, įveskite *Ilgį* ir *Tikslumą*.
4. Spauskite mygtuką *Pridėti į laukų sąrašą*
5. Kartokite aukščiau pateiktus veiksmus visiems pridedamiems laukams
6. Vėliau galite keisti laukų rikiuotę naudodami mygtukus Perkelti aukštyn ir Perkelti žemyn
7. Kai jums jau tinka atributai, spauskite *Gerai*. QGIS pridės naują sluoksnį į legendą ir jūs galėsite redaguoti jį kaip aprašyta skyriuje *Digitizing an existing layer*.

Pagal nutylėjimą pirmas pridedamas sveiko skaičiaus stulpelis `id`, bet jį galima išimti.

Naujo SpatiaLite sluoksnio kūrimas

Norėdami sukurti naują SpatiaLite sluoksnį, spauskite mygtuką *Naujas SpatiaLite sluoksnis…*, kurį rasite meniu *Sluoksnis* ► *Kurti sluoksnį* ► arba įrankinėje *Duomenų šaltinių tvarkyklė*. Jums bus parodytas dialogas *Naujas SpatiaLite sluoksnis*, kaip pavaizduota Fig. 11.28.



11.28 Fig. : Naujo SpatiaLite sluoksnio kūrimo dialogas

1. Pirmas žingsnis yra nurodyti duomenų bazės failo vietą. Tai galima padaryti paspaudus mygtuką ..., kuris yra lauko *Duombazė* dešinėje ir parinkus esamą SpatiaLite failą arba sukūrus naują. QGIS automatiškai pridės reikiamą plėtinį prie jūsų nurodyto pavadinimo.
2. Nurodykite naujo sluoksnio (*Sluoksnio pavadinimą*)

3. Apibrėžkite *Geometrijos tipą*. Jei tai nėra sluoksnis be geometrijos, jūs galite nurodyti, ar jame turi reikia *Įtraukti Z matmenį* ir/ar *Įtraukti M reikšmes*.
4. Naudodami mygtuką nurodykite koordinačių atskaitos sistemą.

Norėdami pridėti laukus į jūsų kuriamą sluoksnį:

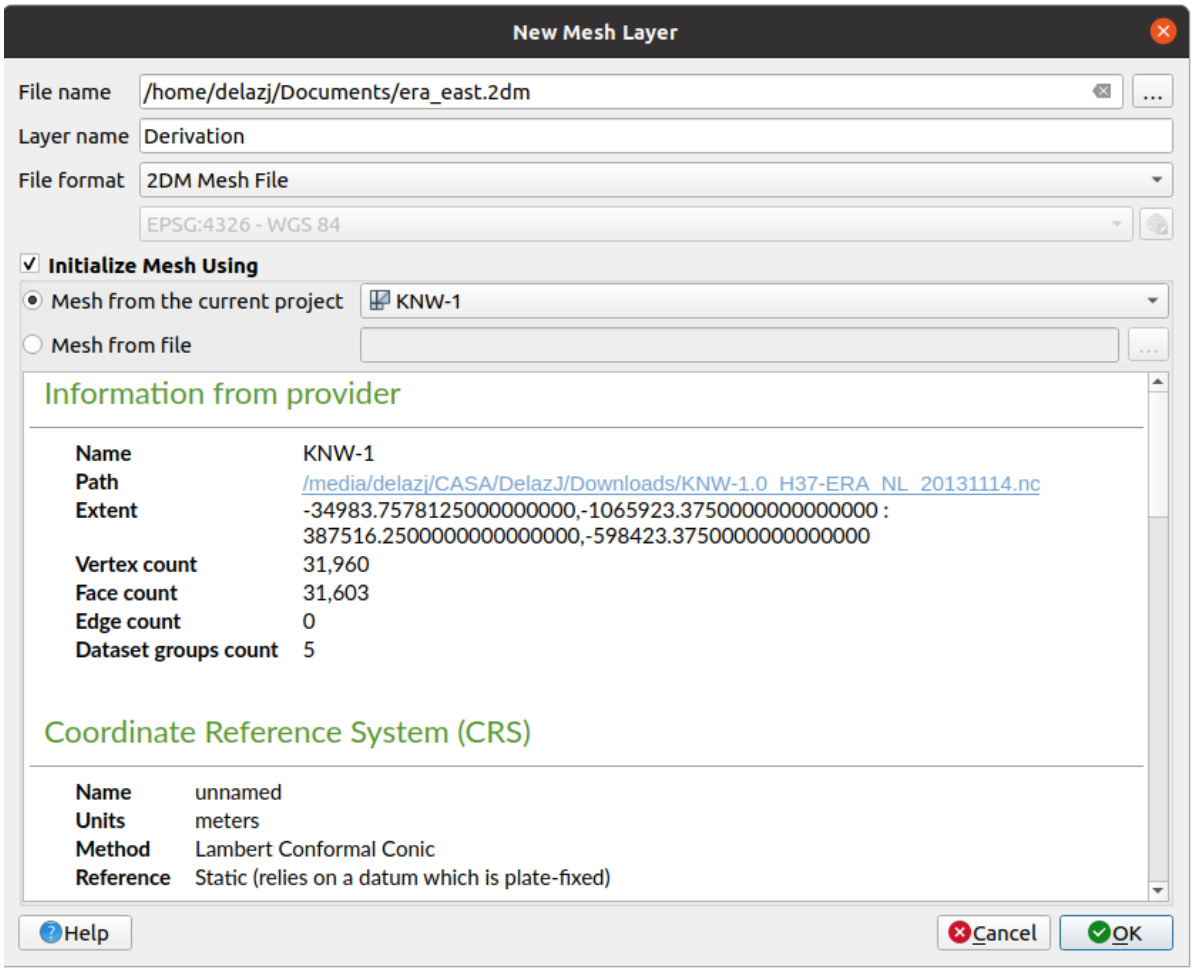
1. Įveskite lauko *Pavadinimą*
2. Parinkite duomenų *Tipą*. Palaikomi tipai yra *Tekstiniai duomenys*, *Sveiki skaičiai*, *Dešimtainiai skaičiai*, *Datos* ir *Datos su laiku*.
3. Spauskite mygtuką *Pridėti į laukų sąrašą*
4. Kartokite aukščiau pateiktus veiksmus visiems pridedamiems laukams
5. Vėliau galite keisti laukų rikiuotę naudodami mygtukus Perkelti aukštyn ir Perkelti žemyn
6. Kai jums jau tinka atributai, spauskite *Gerai*. QGIS pridės naują sluoksnį į legendą ir jūs galėsite redaguoti jį kaip aprašyta skyriuje *Digitizing an existing layer*.

Jei reikia, jūs galite parinkti *Kurti automatiškai didėjantį pirminį raktą* skiltyje *Sudėtingesnės parinktys*. Jūs taipogi galite pervadinti *Geometrijos stulpelį* (pagal nutylėjimą `geometry`).

Tolimesnį SpatiaLite sluoksnių tvarkymą galima atlikti su *DB tvarkykle*.

Creating a new Mesh layer

To create a new Mesh layer, press the *New Mesh Layer…* button in the *Layer* ► *Create Layer* ► menu or from the *Data Source Manager* toolbar. The *New Mesh Layer* dialog will be displayed as shown in Fig. 11.29.



11.29 Fig. : Creating a New Mesh layer dialog

1. The first step is to indicate the mesh file location. This can be done by pressing the … button to the right of the *File name* field and select an existing mesh file or create a new one.
2. Provide a name (*Layer name*), i.e. the name the layer is displayed with in the *Layers* panel
3. Select the *File format*: currently supported mesh file formats are `2DM Mesh File (*.2dm)`, `Selafin File (*.slf)` and `UGRID (*.nc)`.
4. Indicate the *Coordinate Reference System* to assign to the dataset
5. The above steps will generate an empty layer that you can afterwards digitize vertices and add dataset groups to. It's however also possible to initialize the layer with an existing mesh layer, i.e. populate the new layer with vertices or faces from the other. To do so:

    1. Check *Initialize Mesh using*
    2. and select either a *Mesh from the current project* or *Mesh from a file*. Informations on the selected mesh file are displayed for checkup.

    Note that only the frame of the mesh layer is transferred to the new layer; their datasets are not copied.

Creating a new GPX layer

To create a new GPX file:

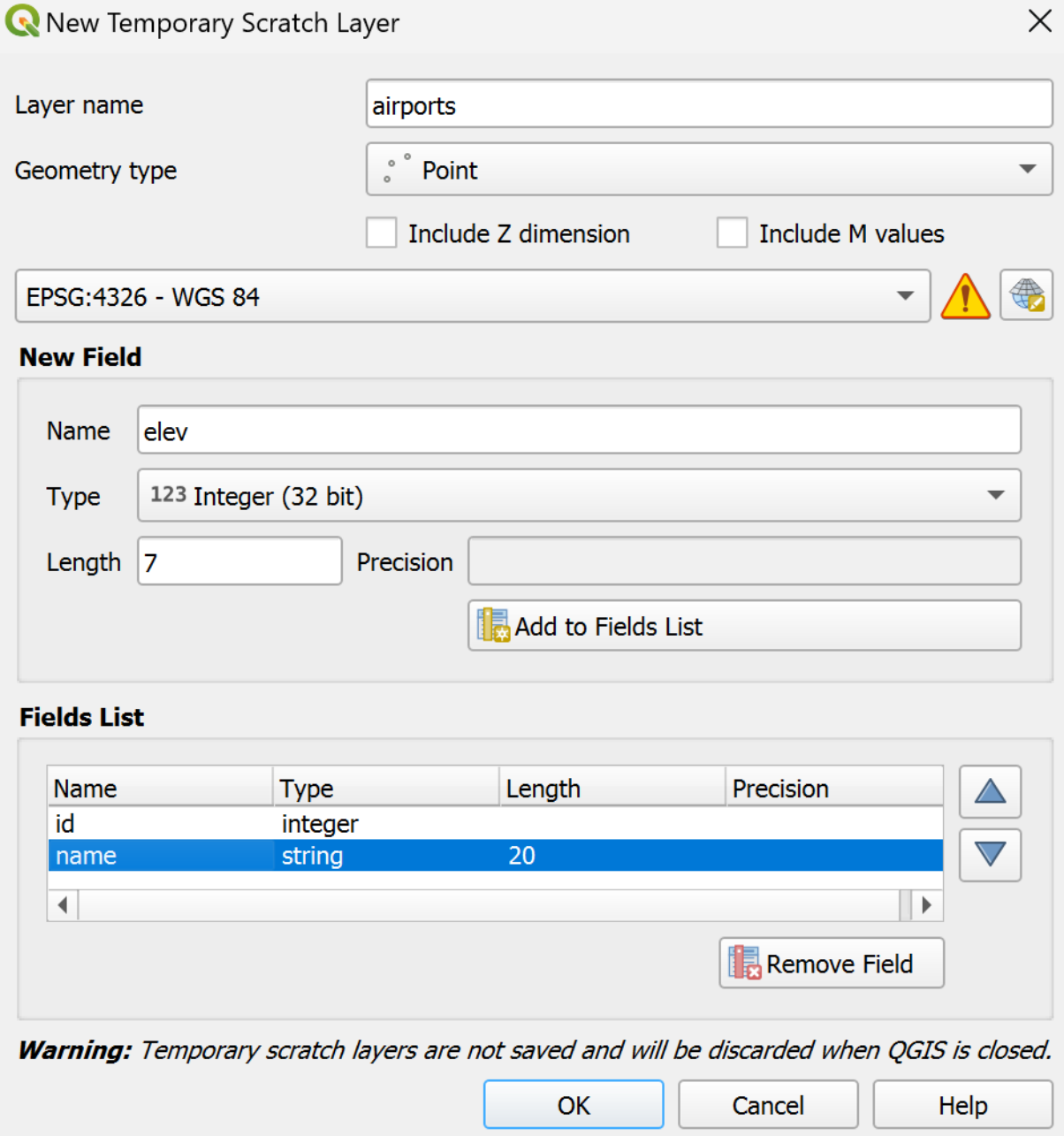
1. Select *Create Layer* ► *New GPX Layer…* from the *Layer* menu.
2. In the dialog, choose where to save the new file, name it and press *Save*.
3. Three new layers are added to the *Layers Panel*:
   - a point layer to digitize locations (`waypoints`) with fields storing the name, elevation, comment, description, source, url and url name
   - a line layer to digitize sequences of locations that make up a planned route (`routes`) with fields storing the name, symbol, number, comment, description, source, url, url name
   - and a line layer to track the receiver's movement over time (`tracks`) with fields storing the name, symbol, number, comment, description, source, url, url name.
4. You can now edit any of them as described in section *Digitizing an existing layer*.

Creating a new Temporary Scratch Layer

Temporary Scratch Layers are in-memory layers, meaning that they are not saved on disk and will be discarded when QGIS is closed. They can be handy for storing features you temporarily need or as intermediate layers during geoprocessing operations.

To create a new Temporary Scratch layer, choose the *New Temporary Scratch Layer…* entry in the *Layer* ► *Create Layer* ► menu or in the *Data Source Manager* toolbar. The *New Temporary Scratch Layer* dialog will be displayed as shown in Fig. 11.30. Then:

1. Provide the *Layer name*
2. Select the *Geometry type*. Here you can create a:
   - `No geometry` type layer, served as simple table,
   - `Point` or `MultiPoint` layer,
   - `LineString/CompoundCurve` or `MultiLineString/MultiCurve` layer,
   - `Polygon/CurvePolygon` or `MultiPolygon/MultiSurface` layer.
3. For geometric types, specify the dimensions of the dataset: check whether it should *Include Z dimension* and/or *Include M values*
4. Naudodami mygtuką nurodykite koordinačių atskaitos sistemą.
5. Add fields to the layer. Note that unlike many formats, temporary layers can be created without any fields. This step is thus optional.
   1. Įveskite lauko *Pavadinimą*
   2. Select the data *Type*: *Text*, *Whole number*, *Decimal number*, *Boolean*, *Date*, *Time*, *Date & Time* and *Binary (BLOB)* are supported.
   3. Depending on the selected data format, enter the *Length* and *Precision*
   4. Spauskite mygtuką *Pridėti į laukų sąrašą*
   5. Repeat the steps above for each field you need to add
   6. Vėliau galite keisti laukų rikiuotę naudodami mygtukus Perkelti aukštyn ir Perkelti žemyn
6. Once you are happy with the settings, click *OK*. QGIS will add the new layer to the *Layers* panel, and you can edit it as described in section *Digitizing an existing layer*.



11.30 Fig. : Creating a new Temporary Scratch layer dialog

You can also create prepopulated temporary scratch layers using e.g. the clipboard (see *Creating new layers from the clipboard*) or as a result of a *Processing algorithm*.

Patarimas

Permanently store a memory layer on disk

To avoid data loss when closing a project with temporary scratch layers, you can save these layers to any vector format supported by QGIS:

- clicking the indicator icon next to the layer;
- selecting the *Make permanent* entry in the layer contextual menu;
- using the *Export* ► entry from the contextual menu or the *Layer* ► *Save As...* menu.

Each of these commands opens the *Save Vector Layer as* dialog described in the *Creating new layers from an existing layer* section and the saved file replaces the temporary one in the *Layers* panel.

11.2.2 Creating new layers from an existing layer

Layers (raster, vector and point cloud) can be saved in a different format and/or reprojected to a different coordinate reference system (CRS) using the *Layer* ► *Save As…* menu or right-clicking on the layer in the *Layers panel* and selecting:

- *Export* ► *Save As…* for raster and point cloud layers
- *Export* ► *Save Features As…* or *Export* ► *Save Selected Features As…* for vector layers.
- Drag and drop the layer from the layer tree to the PostgreSQL entry in the *Browser Panel*. Note that you must have a PostgreSQL connection in the *Browser Panel*.

Bendri parametrai

The *Save Layer as…* dialog shows several parameters to change the behavior when saving the layer. Among the common parameters for raster and vector are:

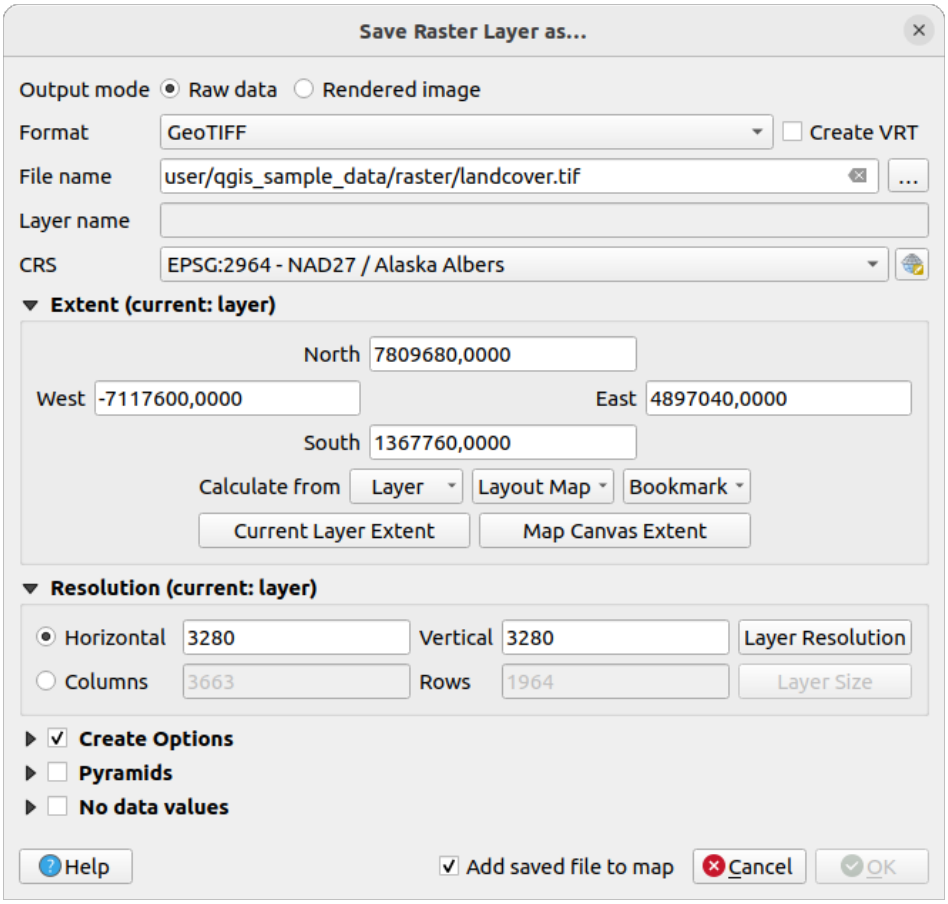
- *File name*: the location of the file on the disk. It can refer to the output layer or to a container that stores the layer (for example database-like formats such as GeoPackage, SpatiaLite or Open Document Spreadsheets).
- *CRS*: can be changed to reproject the data
- *Extent*: restricts the extent of the input that is to be exported using the *extent_selector* widget
- *Add saved file to map*: to add the new layer to the canvas

However, some parameters are specific to certain formats:

Raster specific parameters

Depending on the format of export, some of these options may not be available:

- *Output mode* (it can be **raw data** or **rendered image**)
- *Format*: exports to any raster format GDAL can write to, such as GeoTiff, GeoPackage, MBTiles, Geospatial PDF, SAGA GIS Binary Grid, Intergraph Raster, ESRI .hdr Labelled…
- *Resolution*
- *Create Options*: use advanced options (file compression, block sizes, colorimetry…) when generating files, either from the *predefined create profiles* related to the output format or by setting each parameter.
- *Pyramids* creation
- *VRT Tiles* in case you opted to *Create VRT*
- *No data values*



11.31 Fig. : Saving as a new raster layer

Vector specific parameters

Depending on the format of export, some of these options may be available:

- *Format*: exports to any vector format GDAL can write to, such as GeoPackage, GML, ESRI Shapefile, AutoCAD DXF, ESRI FileGDB, Mapinfo TAB or MIF, SpatiaLite, CSV, KML, ODS, …
- *Layer name*: available when the *File name* refers to a container-like format, this entry represents the output layer.
- *Encoding*
- *Save only selected features*
- *Select fields to export and their export options*: provides means to export fields with custom names and *form widget* settings:
  - Check rows under the *Name* column to choose fields to keep in the output layer, or press *Select All* or *Deselect All* buttons
  - Toggle the *Use aliases for exported name* checkbox to populate the *Export name* column with corresponding field aliases or reset to the original field name. Double-clicking a cell will also edit the name.
  - Depending on whether attribute form custom widgets are in use, you can *Replace all selected raw field values by displayed values*. E.g. if a `value map` widget is applied to a field, the output layer will contain the description values instead of the original values. The replacement can also be done on a field by field basis, in the *Replace with displayed values* column.
- *Persist layer metadata*: ensures that any layer *metadata* present in the source layer will be copied and stored:

  - in the newly created layer, if the output is of GeoPackage format
  - as a `.qmd` file along with the output layer, for other formats. Note that file-based formats supporting more than one dataset (e.g. SpatiaLite, DXF,…) may have unintended behavior.
- *Symbology export*: can be used mainly for DXF export and for all file formats who manage OGR feature styles (see note below) as DXF, KML, tab file formats:
  - **No symbology**: default style of the application that reads the data
  - **Feature symbology**: save style with OGR Feature Styles (see note below)
  - **Symbol Layer symbology**: save with OGR Feature Styles (see note below) but export the same geometry multiple times if there are multiple symbology symbol layers used
  - A **Scale** value can be applied to the latest options

Pastaba

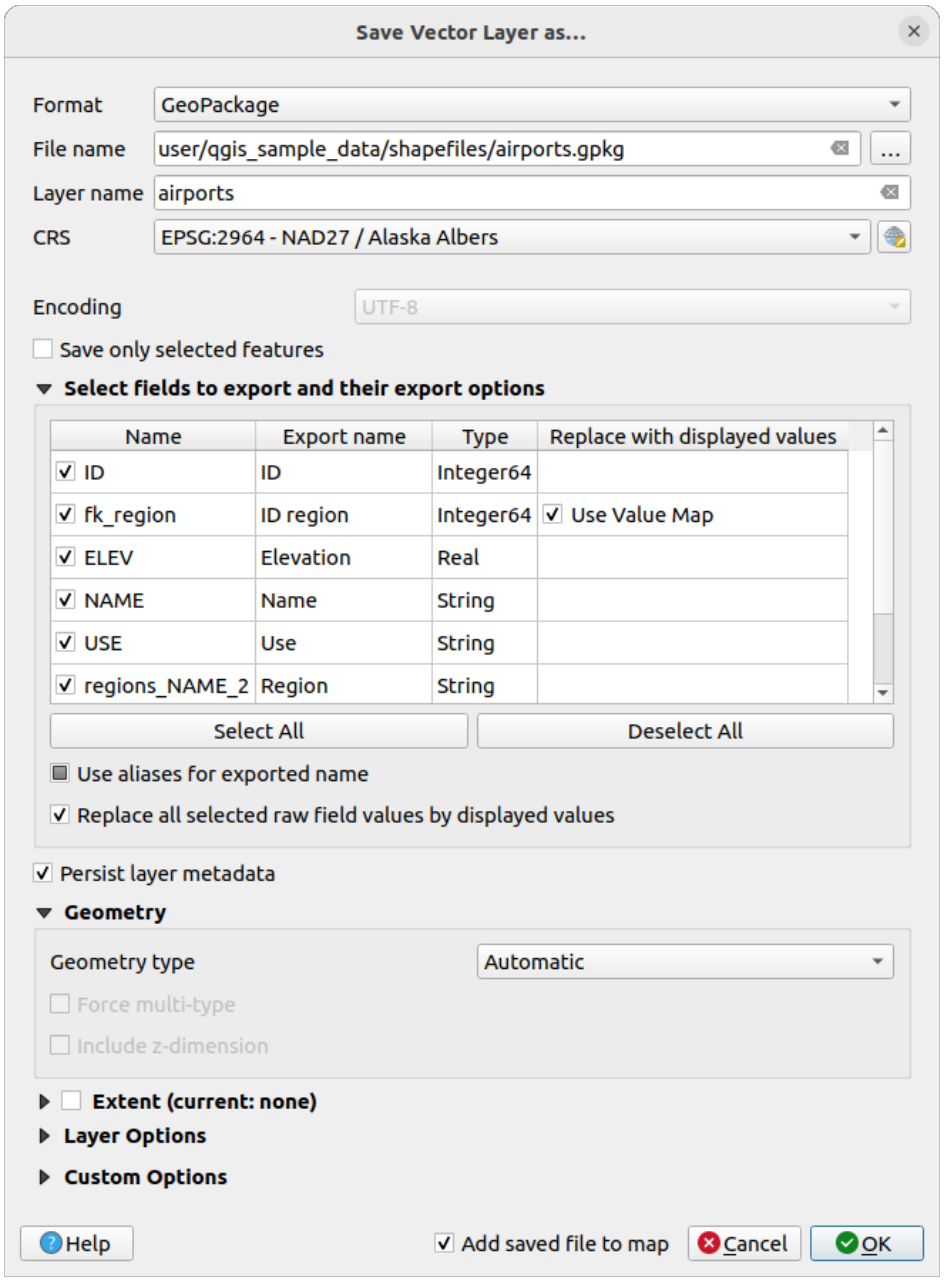
OGR Feature Styles are a way to store style directly in the data as a hidden attribute. Only some formats can handle this kind of information. KML, DXF and TAB file formats are such formats. For advanced details, you can read the OGR Feature Styles specification document.

- *Geometry*: you can configure the geometry capabilities of the output layer
  - *geometry type*: keeps the original geometry of the features when set to **Automatic**, otherwise removes or overrides it with any type. You can add an empty geometry column to an attribute table and remove the geometry column of a spatial layer.
  - *Force multi-type*: forces creation of multi-geometry features in the layer.
  - *Include z-dimension* to geometries.

Patarimas

Overriding layer geometry type makes it possible to do things like save a geometryless table (e.g. `.csv` file) into a shapefile WITH any type of geometry (point, line, polygon), so that geometries can then be manually added to rows with the Add Part tool.

- *Datasource Options*, *Layer Options* or *Custom Options* which allow you to configure advanced parameters depending on the output format. Some are described in *Exploring Data Formats and Fields* but for full details, see the GDAL driver documentation. Each file format has its own custom parameters, e.g. for the `GeoJSON` format have a look at the GDAL GeoJSON documentation.



11.32 Fig. : Saving as a new vector layer

When saving a vector layer into an existing file, depending on the capabilities of the output format (Geopackage, SpatiaLite, FileGDB…), the user can decide whether to:

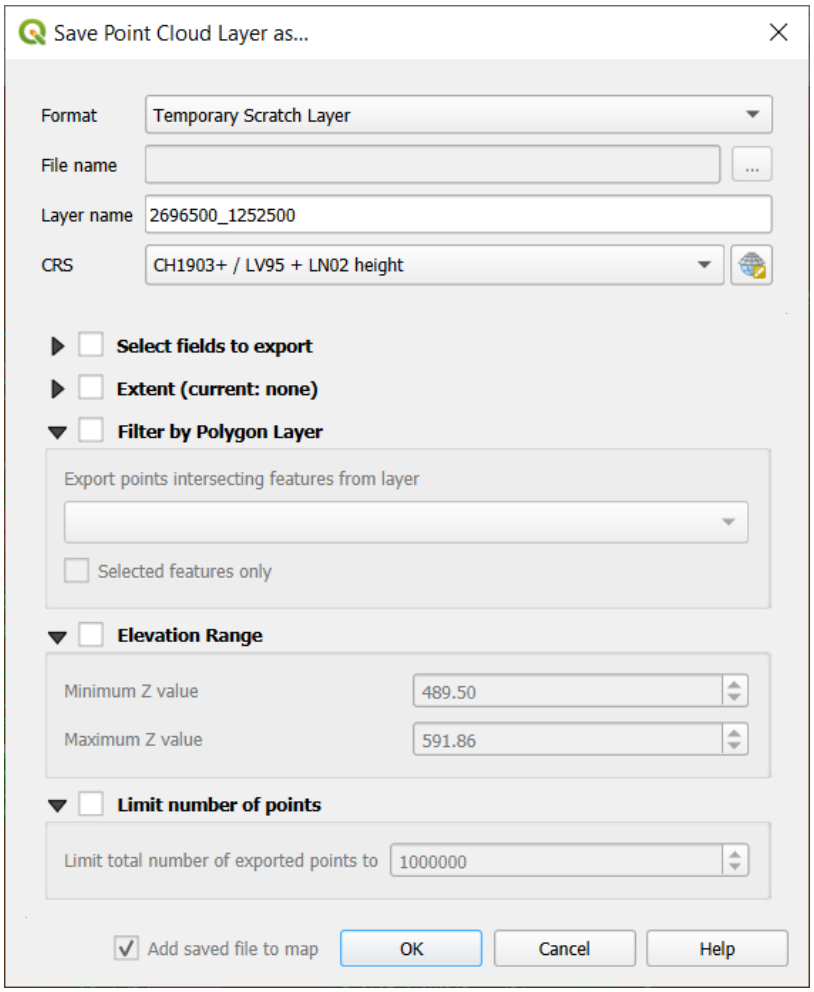
- overwrite the whole file
- overwrite only the target layer (the layer name is configurable)
- append features to the existing target layer
- append features, add new fields if there are any.

For formats like ESRI Shapefile, MapInfo .tab, feature append is also available.

Point Cloud specific parameters

Similar to raster and vector layers, point cloud layers can be saved in a different format and/or reprojected to a different coordinate reference system (CRS). This allows you to export a point cloud layer to vector or point cloud formats. Current supported formats are: Temporary scratch (memory layer), GeoPackage, ESRI Shapefile, DXF and LAS/LAZ point cloud. In addition to the common parameters listed above, exporting point cloud layers includes the following options:

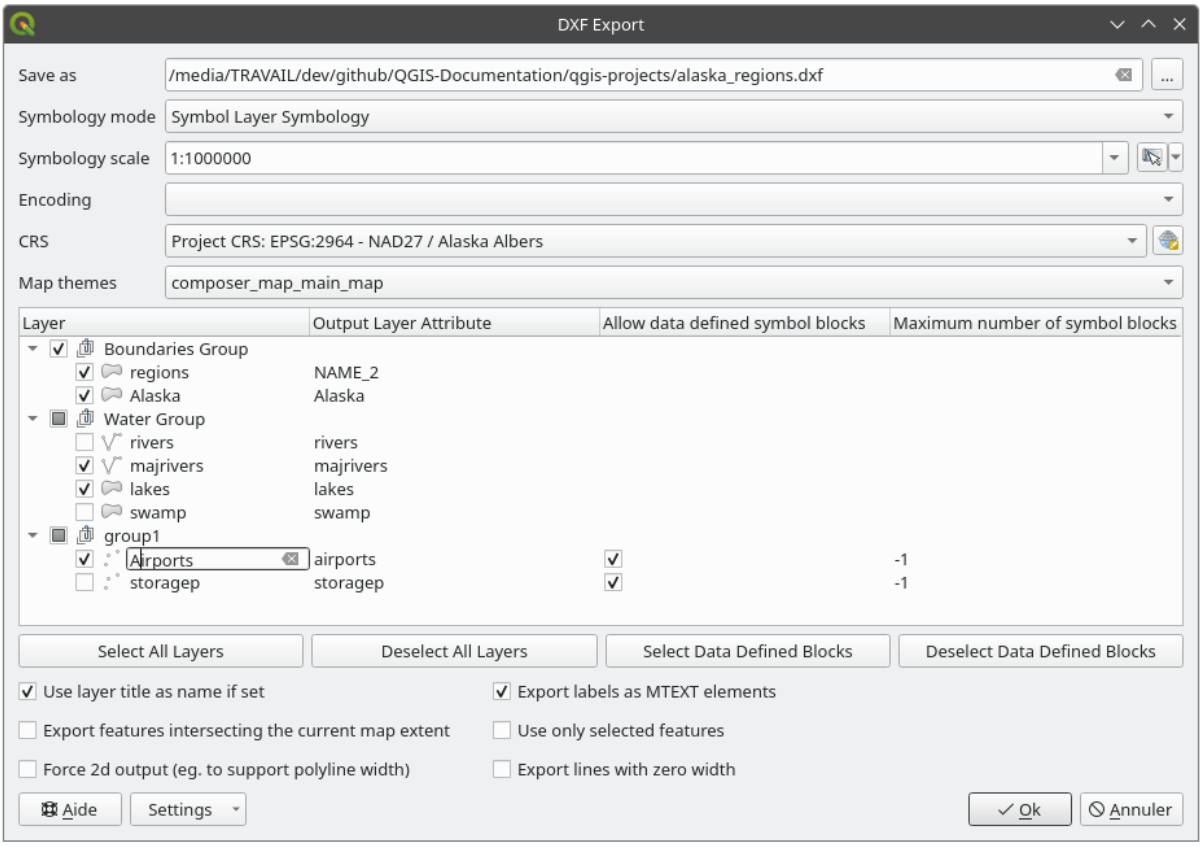
- *Filter by Polygon Layer*: Allows you to filter the point cloud data based on a polygon layer.
- *Elevation Range*: Enables filtering of the point cloud data based on a specified Z range.
- *Limit number of points*: Provides an option to limit the number of points exported from the point cloud layer.



11.33 Fig. : Saving a point cloud layer as a new layer

11.2.3 Creating new DXF files

Besides the *Save As...* dialog which provides options to export a single layer to another format, including `*.DXF`, QGIS provides another tool to export multiple layers as a single DXF layer. It's accessible in the *Project* ► *Import/Export* ► *Export Project to DXF...* menu.



11.34 Fig. : Exporting a project to DXF dialog

In the *DXF Export* dialog:

1. Provide the destination file.
2. Choose the symbology mode and scale (see the *OGR Feature Styles* note), if applicable.
3. Select the data *Encoding*.
4. Select the *CRS* to apply: the selected layers will be reprojected to the given CRS.
5. Select the layers to include in the DXF files either by checking them in the table widget or automatically picking them from an existing *map theme*. The *Select All* and *Deselect All* buttons can help to quickly set the data to export.

   For each layer, you can:

   - Override the output layer name without altering the original project layer. For this, click on the *Layer* name in the dialog and write the output name to use.
   - *Output layer attribute*: Choose whether to export all the features in a single DXF layer or rely on a field whose values are used to split the features into layers in the DXF output. In the latter case, each layer will take its name from the corresponding field value.
   - *Allow data defined symbol blocks*:

- *Maximum number of symbol blocks*: creates symbol blocks up to the specified limit, starting with the ones containing the highest number of references. The other symbols are written as they are. `-1` means no limitation.

Optionally, you can also choose to:

- ☐ *Use the layer title as name if set* instead of the layer name itself: the title is taken from the *metadata* or *server* properties of the layer;
- ☐ *Export features intersecting the current map extent*;
- ☐ *Force 2d output (eg. to support polyline width)*;
- ☑ *Export label as MTEXT elements* or TEXT elements;
- ☐ *Use only selected features*;
- ☐ *Export lines with zero width*: all the lines are exported with minimal width `0` (hairline) if enabled. This helps keep the lines minimal in the file regardless of the zoom level, and can be handy for doing further CAD-editing with the exported dxf, especially if there are many features next to each other on the map.

Pastaba

The precedence for defining the output layer name is as follows:

1. The field value from *Output layer attribute*
2. The overridden name in *Layer* column
3. The *Use the layer title as name if set* option
4. The layer name

Current settings defined in the *DXF Export* dialog may be stored in an XML file for reusing them in other sessions. For this, the *Settings* combo box has two options: *Load Settings from File…* and *Save Settings to File…*.

11.2.4 Creating new layers from the clipboard

Features that are on the clipboard can be pasted into a new layer. To do this, Select some features, copy them to the clipboard, and then paste them into a new layer using *Edit* ► *Paste Features as* ► and choosing:

- *New Vector Layer…*: the *Save vector layer as…* dialog appears (see *Creating new layers from an existing layer* for parameters)
- or *Temporary Scratch Layer…*: you need to provide a name for the layer

A new layer, filled with selected features and their attributes is created (and added to map canvas).

Pastaba

Creating layers from the clipboard is possible with features selected and copied within QGIS as well as features from another application, as long as their geometries are defined using well-known text (WKT).

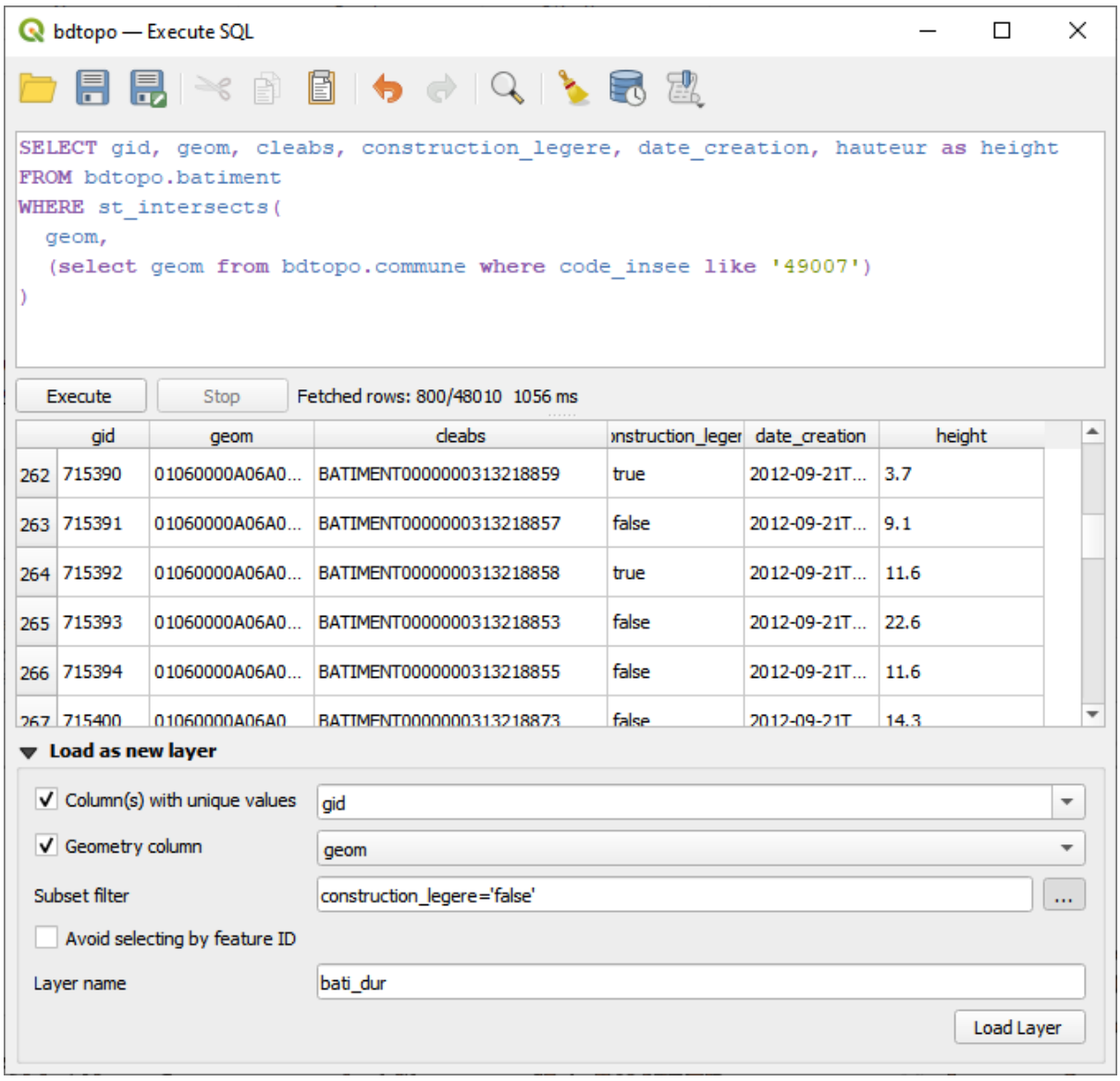
11.2.5 Creating SQL Query Layers

Beside loading an entire layer in a project or creating new layers from scratch or pasted features, you can also load layers generated on the fly from other layer(s). They are the result of a more or less advanced filter using SQL language, applied to stored layers regardless of their data provider or their availability in the active project. This means that temporary layers are not compatible with this feature. Depending on the provider, one or more layers can be used to write the query. The generated layer remains dependent on the layer(s) involved in the query and is loaded with the Filter icon next to it.

This feature is accessible:

- From the *Browser* panel, right-click on a supported data (plain layer, database connection, schema or table) and select *Execute SQL…* entry in the contextual menu.
- From the *Layers* panel, select a loaded layer, right-click and select *Execute SQL…* entry in the contextual menu.

This opens a window with a central text box widget where you can write SQL queries.



11.35 Fig. : Executing SQL queries in the Execute SQL window

At the top of the dialog, a toolbar provides a set of tools to create, store and manipulate your queries:

- Open Queries…: fills the text editor widget with contents from an existing `.sql` file
- Save Queries… and Save Queries as… help you store the written query to a `.sql` file
- Statements of the query can be adjusted using the Cut, Copy and Paste buttons. Likewise, you can Undo or Redo your changes.
- The Find & Replace enables, at the bottom of the dialog, a widget allowing to look for a particular string in your SQL code. The search can be case sensitive, affect partial or whole word, rely on a regular expression. It is then possible to navigate through the found strings, replacing them one by one or all in a row.
- Use Clear SQL Editor to wipe the text editor.
- The History button opens a dialog storing previously run queries. More at *Query History*.
- As previously mentioned, queries can be saved as an `.sql` file stored on disk. Using the Store Current Query button, they can also be stored:
  - In the active *User Profile*, in the associated `QGIS3.ini` file, thus accessible in subsequent projects
  - or as part of the *Current Project*.

  Clicking an entry from the stored queries drop-down menu inserts that query in the expression being written. The stored entry can also be deleted, from the drop-down menu.

Running and loading queries as layer

In the central part of the *Execute SQL* dialog, you build your query using the SQL syntax supported by the underlying provider (e.g., OGR, GeoPackage, PostgreSQL).

By default, if opened from a layer entry, a sample SQL query is provided. Editing tools to select, cut, copy, paste, undo and redo are available as well from the contextual menu.

Patarimas

Finding the right SQL syntax for your dataset

To find the appropriate SQL dialect for your dataset, press the … button next to the *Subset filter* option in the bottom part of the dialog.

When ready, pressing the *Execute* button below the text area will run the query. It is possible to highlight a portion of the SQL to only execute that portion when pressing `Ctrl+R` or clicking the *Execute selection* button. Use the *Stop* button to abort the execution.

A successful query execution will display a table at the bottom of the dialog with returned features. You can select specific cells in the result set. Use the `Ctrl+C` shortcut to copy the selected cells to the clipboard. The copied data is available as a formatted table. This allows you to paste the data into other applications, such as spreadsheet where it will show up as a table.

The returned table can be loaded in QGIS expanding the *Load as new layer* group and configuring parameters (their availability depends on the layer provider):

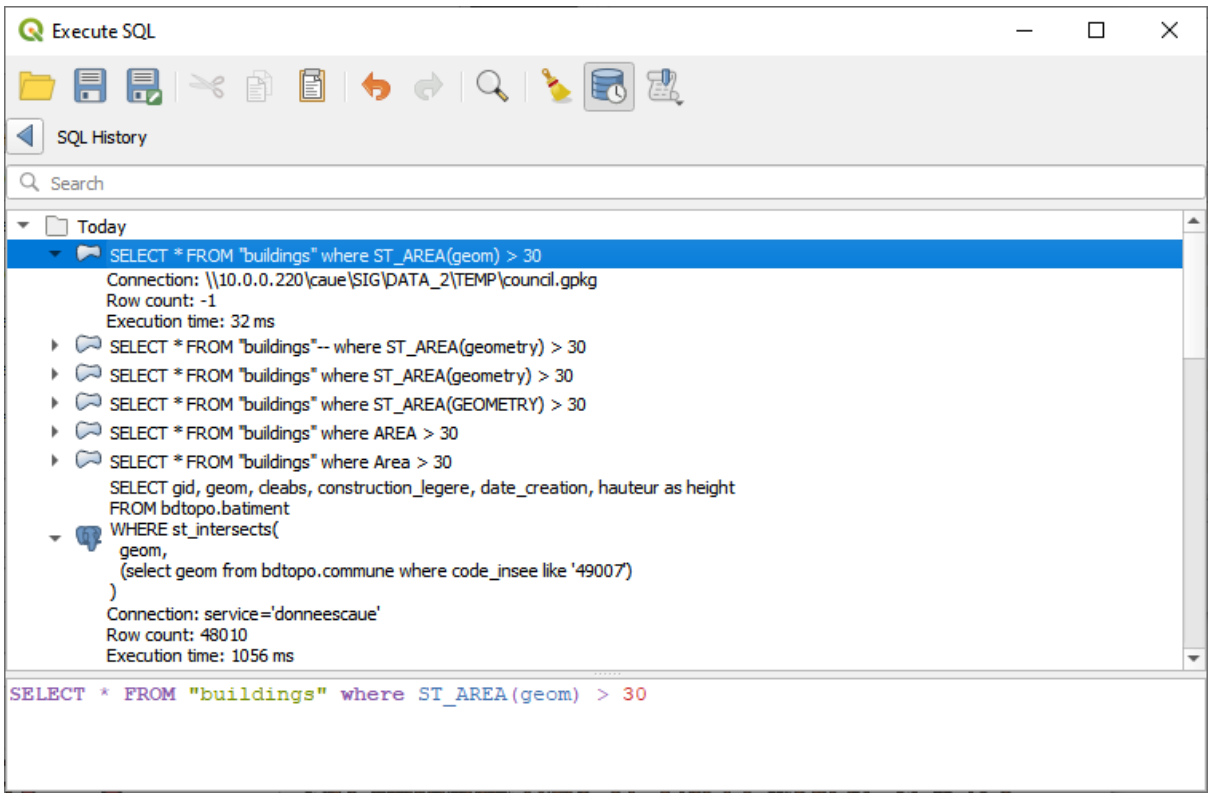
- *Column(s) with unique values* to indicate primary key of the data,
- *Geometry column*: check the box to load the layer as a spatial one, and indicate the geometry field name
- *Subset filter*: allows to filter the results using a `WHERE` clause. It can be written in the text box, or built using the *Query builder* after you pressed …. Make sure to use fields that are available in the SQL layer.
- *Avoid selecting by feature ID*

- *Layer name* in the project

At any time, in the *Layers panel*, you can adjust the output layer by right-clicking and selecting *Update SQL expression…*. The *Update SQL* dialog opens, prefilled with the applied query that you can edit as you wish. Once ready, press *Update layer* and the layer will be modified in-place.

Query History

The *Query History* dialog displays all the previously run queries, sorted by date and data provider type. It is accessible from the *Database* ► *Query History…* menu, or pressing the History button from the *Execute SQL* dialog.



11.36 Fig. : History of the executed SQL queries

At the top of the dialog, a search box can be used to filter out the queries. The search is done through every strings displayed in the query history. If the dialog is opened from within the *Execute SQL* dialog, the search box is preceded by the *Execute SQL toolbar*.

Below, collapsible and sorted date entries list their related queries. Each query is identifiable by its SQL command, and when expanded, reveals:

- The *Connection*: the source of the data the query is based on
- The *Row count*: number of returned features
- The *Execution time*

Hover over an entry and a tooltip displays the full query. Likewise, selecting an entry also displayed it in the bottom part of the dialog. You can interact with the text, copying all or part of it.

Double-click the SQL command entry and the *Execute SQL* dialog opens, filled with the clicked query. Any query being written is overwritten. Right-click the SQL command entry and you can either:

- *Load SQL Command…*, loads the target command into the *Execute SQL* dialog, replacing any existing query. It is the same as double clicking the entry.

- *Copy SQL Command* and paste it wherever you want.

11.2.6 Creating virtual layers

A virtual layer is a special kind of vector layer. It allows you to define a layer as the result of an SQL query involving any number of other vector layers that QGIS is able to open. Virtual layers do not carry data by themselves and can be seen as views.

To create a virtual layer, open the virtual layer creation dialog by:

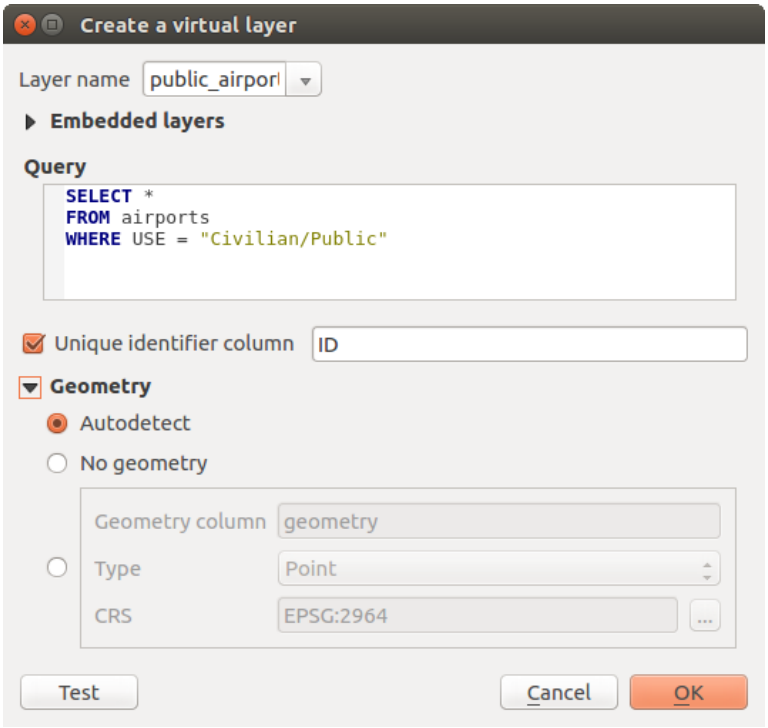
- choosing the *Add/Edit Virtual Layer* entry in the *Layer* ► *Add Layer* ► menu;
- enabling the *Add Virtual Layer* tab in the *Data Source Manager* dialog;
- or using the *DB Manager* dialog tree.

The dialog allows you to specify a *Layer name* and an SQL *Query*. The query can use the name (or id) of loaded vector layers as tables, as well as their field names as columns.

For example, if you have a layer called `airports`, you can create a new virtual layer called `public_airports` with an SQL query like:

```
SELECT *
FROM airports
WHERE USE = "Civilian/Public"
```

The SQL query will be executed, regardless of the underlying provider of the `airports` layer, even if this provider does not directly support SQL queries.



11.37 Fig. : Create virtual layers dialog

Joins and complex queries can also be created, for example, to join airports and country information:

```
SELECT airports.*, country.population
FROM airports
JOIN country
ON airports.country = country.name
```

Pastaba

It's also possible to create virtual layers using the SQL window of *DB Manager Plugin*.

Embedding layers for use in queries

Besides the vector layers available in the map canvas, the user can add layers to the *Embedded layers* list, which can be used in queries without the need to have them showing in the map canvas or Layers panel.

To embed a layer, click *Add* and provide the *Local name*, *Provider*, *Encoding* and the path to the *Source*.

The *Import* button allows adding layers in the map canvas into the Embedded layers list. Those layers can then be removed from the Layers panel without breaking existent queries.

Supported query language

The underlying engine uses SQLite and SpatiaLite to operate.

It means you can use all of the SQL your local installation of SQLite understands.

Functions from SQLite and spatial functions from SpatiaLite can also be used in a virtual layer query. For instance, creating a point layer out of an attribute-only layer can be done with a query similar to:

```
SELECT id, MakePoint(x, y, 4326) as geometry
FROM coordinates
```

Functions of QGIS expressions can also be used in a virtual layer query.

To refer the geometry column of a layer, use the name `geometry`.

Contrary to a pure SQL query, all the fields of a virtual layer query must be named. Don't forget to use the `as` keyword to name your columns if they are the result of a computation or a function call.

Performance issues

With default parameters, the virtual layer engine will try its best to detect the type of the different columns of the query, including the type of the geometry column if one is present.

This is done by introspecting the query when possible or by fetching the first row of the query (LIMIT 1) as a last resort. Fetching the first row of the result just to create the layer may be undesirable for performance reasons.

The creation dialog parameters:

- *Unique identifier column*: specifies a field of the query that represents unique integer values that QGIS can use as row identifiers. By default, an autoincrementing integer value is used. Defining a unique identifier column speeds up the selection of rows by id.
- *No geometry*: forces the virtual layer to ignore any geometry field. The resulting layer is an attribute-only layer.
- Geometry *Column*: specifies the name of the geometry column.
- Geometry *Type*: specifies the type of the geometry.
- Geometry *CRS*: specifies the coordinate reference system of the virtual layer.

Special comments

The virtual layer engine tries to determine the type of each column of the query. If it fails, the first row of the query is fetched to determine column types.

The type of a particular column can be specified directly in the query by using some special comments.

The syntax is the following: `/*:type*/`. It has to be placed just after the name of a column. `type` can be either `int` for integers, `real` for floating point numbers or `text`.

For instance:

```
SELECT id+1 as nid /*:int*/
FROM table
```

The type and coordinate reference system of the geometry column can also be set thanks to special comments with the following syntax `/*:gtype:srid*/` where `gtype` is the geometry type (`point`, `linestring`, `polygon`, `multipoint`, `multilinestring` or `multipolygon`) and `srid` an integer representing the EPSG code of a coordinate reference system.

Use of indexes

When requesting a layer through a virtual layer, the source layer indices will be used in the following ways:

- if an = predicate is used on the primary key column of the layer, the underlying data provider will be asked for a particular id (FilterFid)
- for any other predicates (>, <=, !=, etc.) or on a column without a primary key, a request built from an expression will be used to request the underlying vector data provider. It means indexes may be used on database providers if they exist.

A specific syntax exists to handle spatial predicates in requests and triggers the use of a spatial index: a hidden column named `_search_frame_` exists for each virtual layer. This column can be compared for equality to a bounding box. Example:

```
SELECT *
FROM vtab
WHERE _search_frame_=BuildMbr(-2.10,49.38,-1.3,49.99,4326)
```

Spatial binary predicates like `ST_Intersects` are sped up significantly when used in conjunction with this spatial index syntax.

11.3 Geo-koordinavimas

Geo-koordinatorius yra įrankis, skirtas nekoordinuotų rastro ar vektorinių sluoksnių pririšimui prie žinomų koordinačių sistemų naudojant žemės kontrolės taškus (GCP). Jis leidžia eksportuoti transformuotus rastrus (GeoTIFF) ir vektorius (shape failus, geopackages ir t.t.) ar įrašyti atitinkamus pasaulio failus (tik rastrams). Bazinis sluoksnio geo-koordinavimo būdas yra surasti jo taškus, kuriems jūs galite tiksliai nurodyti koordinates.

Savybės

11.1 Table : Georeferencer Tools

Piktograma	Tikslas	Piktograma	Tikslas
	Atverti rastrą		Atverti vektorių
	Pradėti geo-koordinavimą		Kurti GDAL skriptą
	Įkelti GCP taškus		Įrašyti GCP taškus kaip
	Transformavimo nustatymai		
	Pridėti GCP tašką		Trinti GCP tašką
	Perkelti GCP tašką		Snapping Type
	Show Advanced Digitizing Dock		Pastumti
	Priartinti		Nutolinti
	Priartinti prie sluoksnio		Paskutinis mastelis
	Kitas mastelis		Susieti geo-koordinatorių su QGIS
	Susieti QGIS su geo-koordinatoriumi		Pilna histogramos apimtis
	Vietinė histogramos apimtis		

11.3.1 Rastro sluoksnio geo-koordinavimas

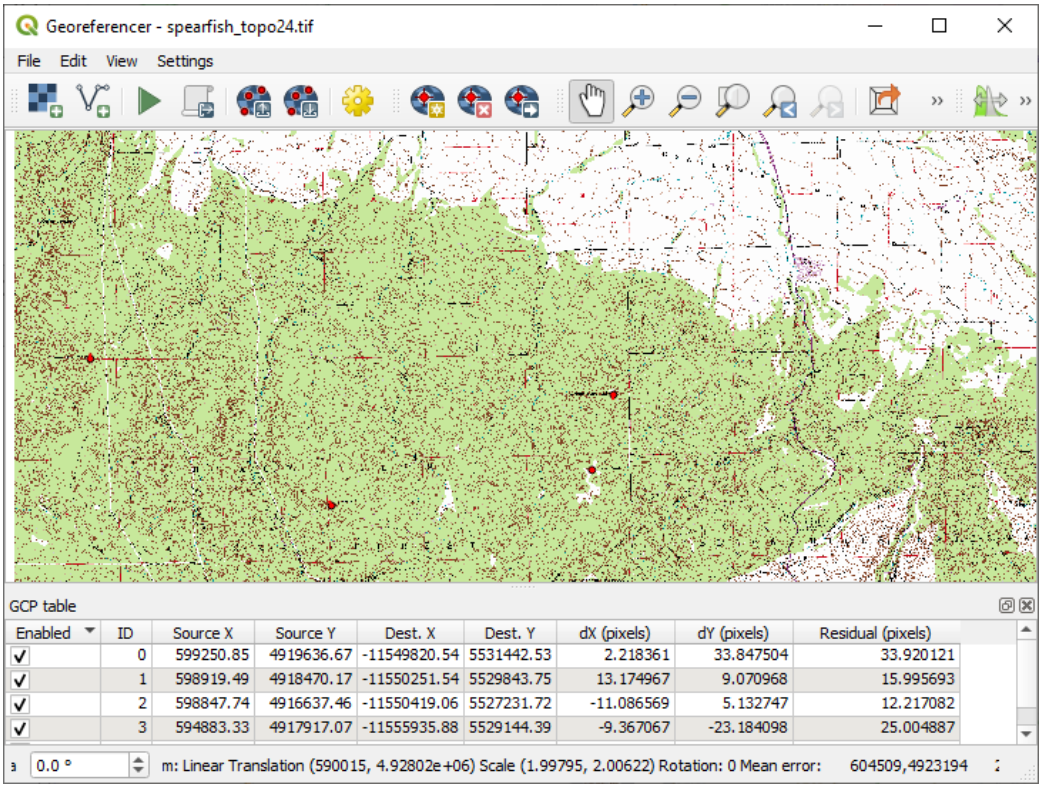
Nurodant X ir Y kordinates (DMS (dd mm ss.ss), DD (dd.dd) ar projektuotos koordinatės (mmmm.mm)), kurios atitinka piešinyje parinktą tašką, galima naudoti vieną iš dviejų būdų:

- Kartais pačiame rastre yra kryžiukai su koordinatėmis „užrašytomis“ ant piešinio. Tokiu atveju jūs galite įvesti koordinates rankiniu būdu.
- Galima naudoti jau geo-koordinuotus sluoksnius. Tai gali būti vektoriniai arba rastro duomenys, kuriuose yra tie patys objektai/geoobjektai, kaip ir piešinyje, kurį jūs norite geo-koordinuoti. Tokiu atveju galite įvesti koordinates paspausdami atskaitos duomenų rinkinį, įkeltą į QGIS žemėlapio drobę.

Įprastinė piešinio geo-koordinavimo procedūra yra parinkti kelis taškus rastre, nurodant jų koordinates, tada parenkant transformacijos tipą. Priklausomai nuo įvesties parametrų ir duomenų, Geo-koordinatorius paskaičiuos pasaulio failo parametrus. Kuo daugiau pateiksite koordinačių, tuo geresnis bus rezultatas.

Pirmas žingsnis yra paleisti QGIS ir spausti *Sluoksnis* ► *Geo-koordinatorius*, kurį jūs rasite QGIS meniu juostoje. Pasirodys geo-koordinatoriaus dialogas, kaip pavaizduota Fig. 11.38.

Šiam pavyzdžiui mes naudojame Šiaurės Dakotos SDGS topo lapą. Jį vėliau galima vizualizuoti kartu su duomenimis iš GRASS vietos `spearfish60`. Galite atsisiųsti topo lapą iš čia: https://grass.osgeo.org/sampledata/spearfish_toposheet.tar.gz.



11.38 Fig. : Geo-koordinavimo dialogas

Žemės kontrolinių taškų (GCP) įvedimas

1. Norėdami pradėti geo-koordinuoti nepriristą rastrą, mes visų pirma turime jį įsikelti mygtuku . Rastras pasirodys pagrindiniame dialogo darbo plote. Įkėlus rastrą mes galime pradėti įvedinėti kontrolinius taškus.

2. Naudodami mygtuką Pridėti GCP tašką, pridėkite taškus į pagrindinį darbo plotą ir įveskite jų koordinates (žr. iliustraciją Fig. 11.39). Tam turite kelis būdus:

   - Spauskite tašką rastro piešyje ir tada įveskite X ir Y koordinates rankiniu būdu, kartu su taško CRS.
   - Spauskite tašką rastro piešinyje ir parinkite mygtuką Iš žemėlapio drobės, kad pridėtumėte X ir Y koordinates, naudodami jau geo-koordinuotą žemėlapį, kuris įkeltas QGIS žemėlapio drobėje. CRS bus nustatyta automatiškai.
   - Įvesdami GCP iš pagrindinės žemėlapio drobės, galite pasirinkti paslėpti geo-koordinatoriaus langą, kol parenkate taškus pagrindinėje drobėje. Jei įjungta parinktis *Automatiškai slėpti geo-koordinatoriaus langą*, paspaudus *Iš žemėlapio drobės*, geo-koordinatoriaus langas bus paslėptas, kol žemėlapio drobėje nepridėsite taško. Dialogas *Įveskite žemėlapio koordinates* liks atvertas. Jei parinktis neįjungta, abu langai liks atidaryti parenkant taškus žemėlapio drobėje. Ši parinktis įsigalioja tik kai geo-koordinavimo langas nėra prisegtas pagrindinėje naudotojo sąsajoje.

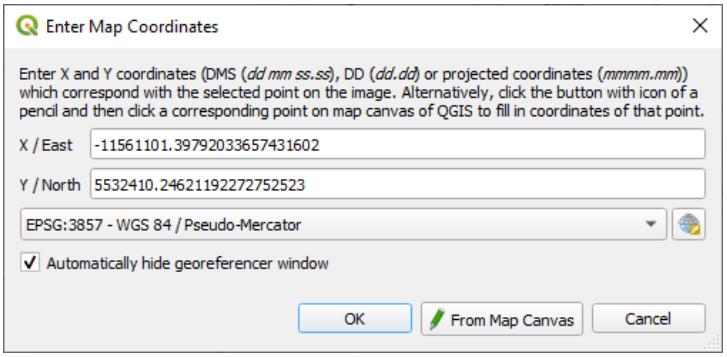
3. Toliau pridėkite taškus. Turėtumėte pridėti bent keturis taškus, kuo daugiau koordinačių galėsite pateikti, tuo geresnis bus rezultatas. Yra papildomi įrankiai darbo ploto priartinimui ar pastūmimui, kad galėtumėte rasti atitinkamus GCP taškus.

> **Patarimas**
>
> Kad išvengtumėte pastovaus persijunginėjimo tarp mygyukų Pastumti, Pridėti GCP tašką ir

> Perkelti GCP tašką, jūs galite naudoti klavišus judėjimui ir pelės ratuką patogiam geo-koordinavimo žemėlapio mastelio keitimui.

4. Kai pateiksite kelis taškus, galite naudoti mygtukus Susieti QGIS su geo-koordinatoriumi ir/ar Susieti geo-koordinatorių su QGIS, kad atitinkamai pakeistumėte pagrindinio QGIS lango vaizdą, kad jis rodytų geo-koordinatoriaus vaizdą arba atvirkščiai.

5. Su įrankiu jūs galite perkelti GCP tiek drobėje, tiek ir geo-koordinavimo lange, jei jums reikia juos patikslinti.

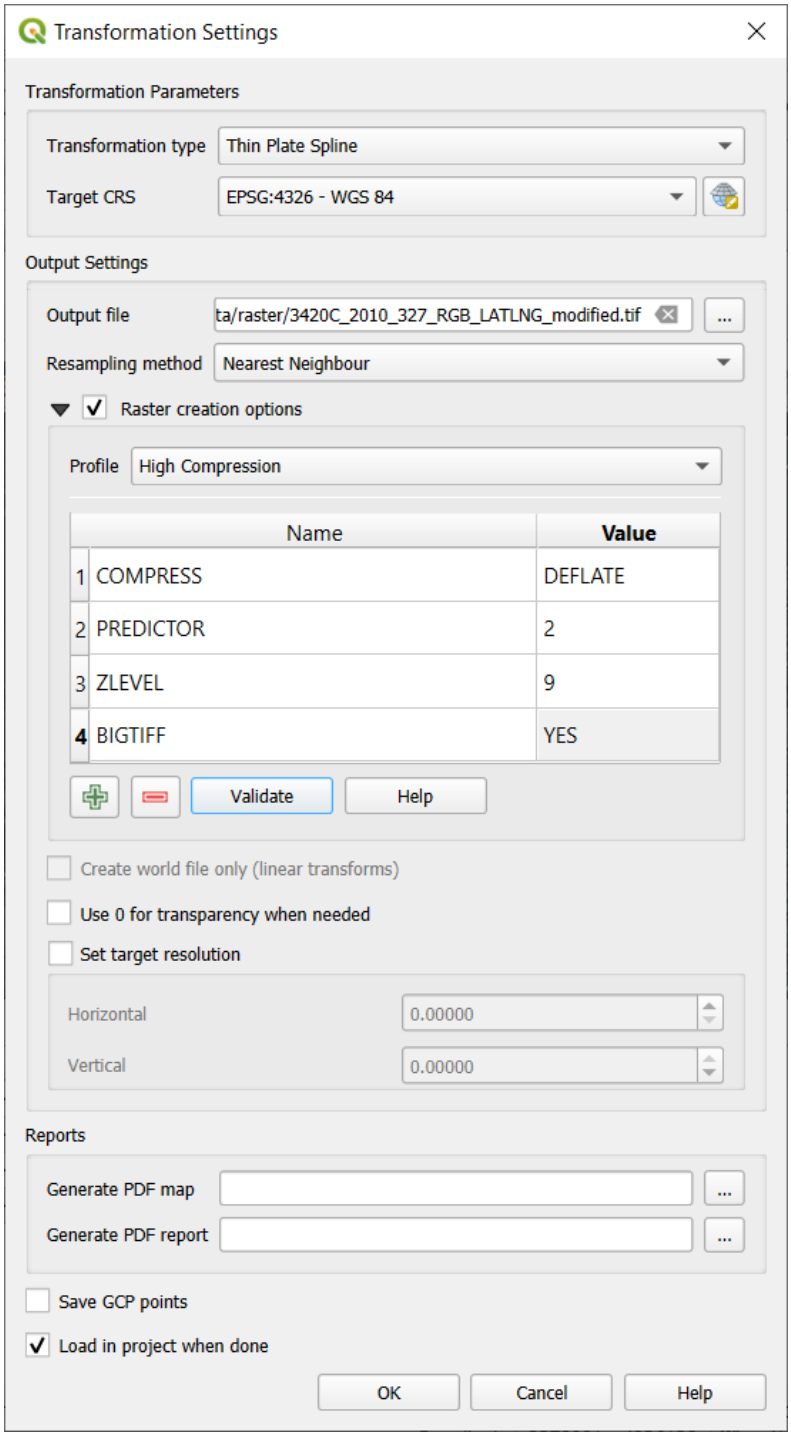


11.39 Fig. : Pridėti taškus rastro piešinyje

Į žemėlapį pridėti taškai bus saugomi atskirame tekstiniame faile (`[failovardas].points`), paprastai kartu su rastro piešiniu. Tai leis mums vėliau iš naujo atidaryti geo-koordinatorių ir pridėti naujus taškus arba ištrinti esamus, kad pagerintumėme rezultatą. Taškų faile yra reikšmės iš: `mapX, mapY, pixelX, pixelY`. Jūs galite naudoti mygtukus Įkelti GCP taškus ir Įrašyti GCP taškus kaip, kad tvarkytumėte šiuos failus.

Defining the transformation settings

After you have added your GCPs to the raster image, you need to define the transformation settings for the georeferencing process.



11.40 Fig. : Defining the georeferencer transformation settings

Available Transformation algorithms

A number of transformation algorithms are available, dependent on the type and quality of input data, the nature and amount of geometric distortion that you are willing to introduce to the final result, and the number of ground control points (GCPs).

Currently, the following *Transformation types* are available:

- The **Linear** algorithm is used to create a world file and is different from the other algorithms, as it does not actually transform the raster pixels. It allows positioning (translating) the image and uniform scaling, but no rotation or other transformations. It is the most suitable if your image is a good quality raster map, in a known CRS, but is just missing georeferencing information. At least 2 GCPs are needed.
- The **Helmert** transformation also allows rotation. It is particularly useful if your raster is a good quality local map or orthorectified aerial image, but not aligned with the grid bearing in your CRS. At least 2 GCPs are needed.
- The **Polynomial 1** algorithm allows a more general affine transformation, in particular also a uniform shear. Straight lines remain straight (i.e., collinear points stay collinear) and parallel lines remain parallel. This is particularly useful for georeferencing data cartograms, which may have been plotted (or data collected) with different ground pixel sizes in different directions. At least 3 GCP's are required.
- The **Polynomial** algorithms 2-3 use more general 2nd or 3rd degree polynomials instead of just affine transformation. This allows them to account for curvature or other systematic warping of the image, for instance photographed maps with curving edges. At least 6 (respectively 10) GCP's are required. Angles and local scale are not preserved or treated uniformly across the image. In particular, straight lines may become curved, and there may be significant distortion introduced at the edges or far from any GCPs arising from extrapolating the data-fitted polynomials too far.
- The **Projective** algorithm generalizes Polynomial 1 in a different way, allowing transformations representing a central projection between 2 non-parallel planes, the image and the map canvas. Straight lines stay straight, but parallelism is not preserved and scale across the image varies consistently with the change in perspective. This transformation type is most useful for georeferencing angled photographs (rather than flat scans) of good quality maps, or oblique aerial images. A minimum of 4 GCPs is required.
- Finally, the **Thin Plate Spline** (TPS) algorithm „rubber sheets" the raster using multiple local polynomials to match the GCPs specified, with overall surface curvature minimized. Areas away from GCPs will be moved around in the output to accommodate the GCP matching, but will otherwise be minimally locally deformed. TPS is most useful for georeferencing damaged, deformed, or otherwise slightly inaccurate maps, or poorly orthorectified aerials. It is also useful for approximately georeferencing and implicitly reprojecting maps with unknown projection type or parameters, but where a regular grid or dense set of ad-hoc GCPs can be matched with a reference map layer. It technically requires a minimum of 10 GCPs, but usually more to be successful.

In all of the algorithms except TPS, if more than the minimum GCPs are specified, parameters will be fitted so that the overall residual error is minimized. This is helpful to minimize the impact of registration errors, i.e. slight imprecisions in pointer clicks or typed coordinates, or other small local image deformations. Absent other GCPs to compensate, such errors or deformations could translate into significant distortions, especially near the edges of the georeferenced image. However, if more than the minimum GCPs are specified, they will match only approximately in the output. In contrast, TPS will precisely match all specified GCPs, but may introduce significant deformations between nearby GCPs with registration errors.

Define the Resampling method

The type of resampling you choose will likely depend on your input data and the ultimate objective of the exercise. If you don't want to change statistics of the raster (other than as implied by nonuniform geometric scaling if using other than the Linear, Helmert, or Polynomial 1 transformations), you might want to choose ‚Nearest neighbour'. In contrast, ‚cubic resampling', for instance, will usually generate a visually smoother result.

It is possible to choose between five different resampling methods:

1. Nearest neighbour
2. Bilinear (2x2 kernel)
3. Cubic (4x4 kernel)
4. Cubic B-Spline (4x4 kernel)
5. Lanczos (6x6 kernel)

Define the Raster creation options

When exporting a raster, *Raster creation options* allows you to define additional options that control how the output file is structured and compressed. See more at *Raster driver options*.

Patarimas

Select an empty entry if you want to create your own custom combination of parameters.

Define the transformation settings

There are several options that need to be defined for the georeferenced output raster.

- The *Create world file* checkbox is only available if you decide to use the linear transformation type, because this means that the raster image actually won't be transformed. In this case, the *Output raster* field is not activated, because only a new world file will be created.
- For all other transformation types, you have to define an *Output raster*. As default, a new file ([filename]_modified) will be created in the same folder together with the original raster image.
- As a next step, you have to define the *Target CRS* (Coordinate Reference System) for the georeferenced raster (see *Darbas su projekcijomis*).
- If you like, you can **generate a pdf map** and also **a pdf report**. The report includes information about the used transformation parameters, an image of the residuals and a list with all GCPs and their RMS errors.
- Furthermore, you can activate the *Set Target Resolution* checkbox and define the pixel resolution of the output raster. Default horizontal and vertical resolution is 1.
- The *Use 0 for transparency when needed* can be activated, if pixels with the value 0 shall be visualized transparent. In our example toposheet, all white areas would be transparent.
- The *Save GCP Points* will store GCP Points in a file next to the output raster.
- Finally, *Load in project when done* loads the output raster automatically into the QGIS map canvas when the transformation is done.

Running the transformation

After all GCPs have been collected and all transformation settings are defined, just press the Start georeferencing button to create the new georeferenced raster.

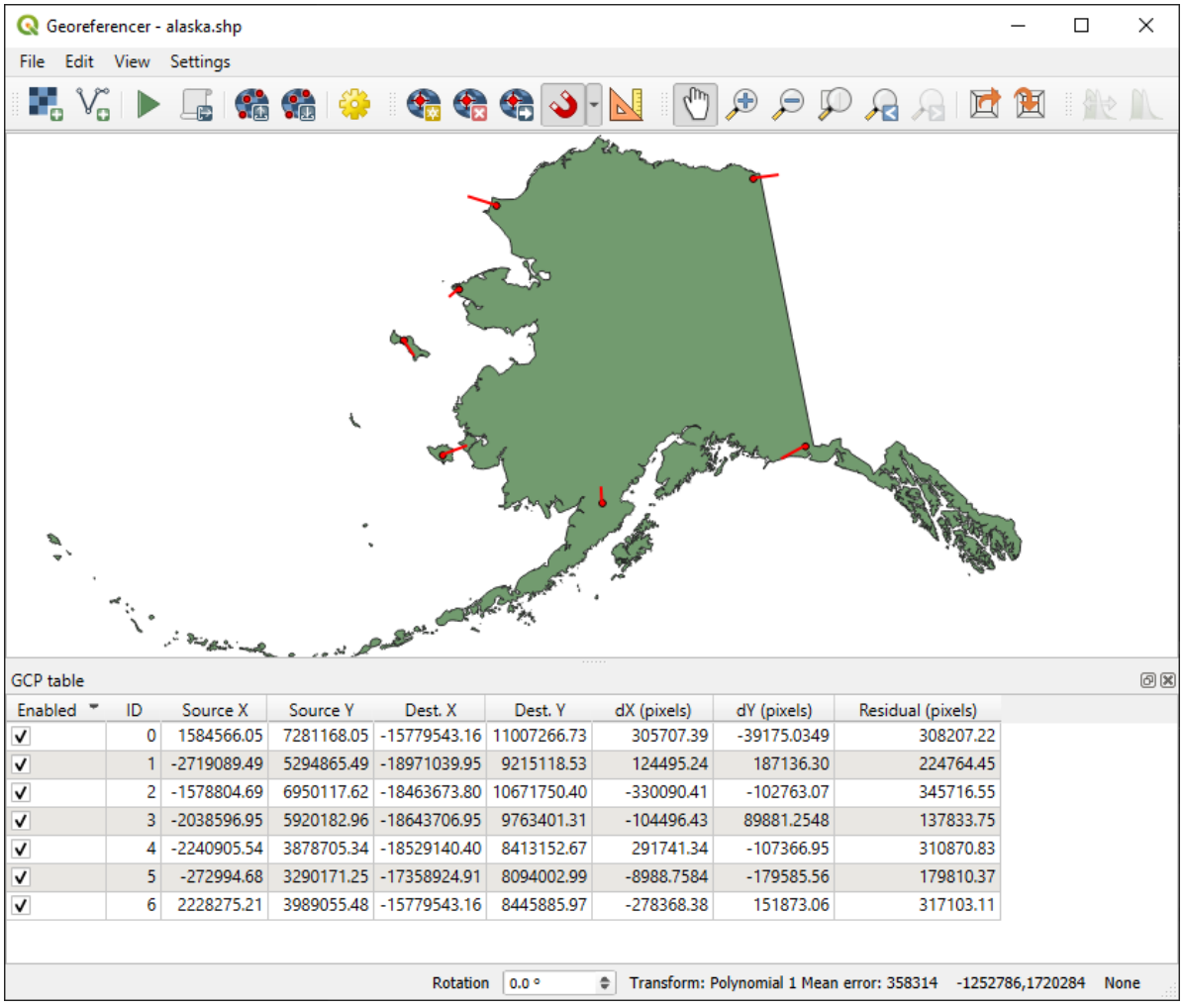
11.3.2 Georeferencing vector layer

Georeferencing vector layers works similarly to raster georeferencing, but instead of matching image pixels, you match vector geometries (points, lines, or polygons) to known spatial references.

The standard procedure starts the same as for raster georeferencing: open QGIS and add a layer to the map canvas to use as a reference. This can be a georeferenced raster or vector layer, or a WMS layer.

Open the Georeferencer dialog from *Layer* ► *Georeferencer*.

Start georeferencing by following these steps (in this example, we use an unreferenced `alaska.shp`):



11.41 Fig. : Vector Georeferencer Dialog

1. Load the unreferenced vector layer using the button. The vector layer will appear in the main working area of the dialog.

2. Activate snapping by clicking the button and selecting the desired snapping type(s). This enables snapping to the reference layer when placing GCPs.

You can also use the Advanced Digitizing dock to ensure high-precision point selection. For more information, refer to the *Advanced Digitizing* section.

3. Use the Add GCP Point button to add a point to the working area. Enter its coordinates manually and set the CRS, or click the From map canvas button to pick the coordinates from a georeferenced layer in the main QGIS map canvas. In that case, the CRS will be set automatically.
4. Define the transformation settings:
   - Select the *Transformation type*. The transformation algorithms are the same as those for raster georeferencing. See *Defining the transformation settings* for more details.
   - Define the *Target CRS* (Coordinate Reference System) for the georeferenced vector (see *Darbas su projekcijomis*).
   - Set the output file format and path (e.g., GeoPackage, Shapefile). By default, a new file with suffix `_modified` will be created in the same folder as the original vector file.
   - Optionally, enable **Generate PDF map** and **Generate PDF report**. The report includes transformation parameters, GCP residuals, and a summary of RMS errors.
   - Enable *Save GCP Points* to store GCPs in a file alongside the output vector layer.
   - Enable *Load in project when done* to add the result directly to the map canvas.
5. Click Start georeferencing to run the transformation and generate the georeferenced vector layer.

11.3.3 Show and adapt layer properties

Clicking on the *Source properties* option in the *Settings* menu opens the *Raster Layer properties* or *Vector Layer properties* depending on the type of layer you are georeferencing.

11.3.4 Configure the georeferencer

You can customize the behavior of the georeferencer in *Settings* ► *Configure Georeferencer* (or use keyboard shortcut `Ctrl+P`).

- Under *Point Tip* you can use the checkboxes to toggle displaying GCP IDs and X/Y coordinates in both the Georeferencer window and the main map canvas.
- *Residual Units* controls whether residual units are given in pixels or map units
- *PDF Report* allows you to set margin size in mm for the report export
- *PDF Map* allows you to choose a paper size for the map export
- Finally, you can activate to *Show Georeferencer window docked*. This will dock the Georeferencer window in the main QGIS window rather than showing it as a separate window that can be minimized.

11.4 Exploring Data Formats and Fields

11.4.1 Raster data

GIS raster data are matrices of discrete cells that represent features / phenomena on, above or below the earth's surface. Each cell in the raster grid has the same size, and cells are usually rectangular (in QGIS they will always be rectangular). Typical raster datasets include remote sensing data, such as aerial photography, or satellite imagery and modelled data, such as elevation or temperature.

Unlike vector data, raster data typically do not have an associated database record for each cell. They are geocoded by pixel resolution and the X/Y coordinate of a corner pixel of the raster layer. This allows QGIS to position the data correctly on the map canvas.

The GeoPackage format is convenient for storing raster data when working with QGIS. The popular and powerful GeoTiff format is a good alternative.

QGIS makes use of georeference information inside the raster layer (e.g., GeoTiff) or an associated *world file* to properly display the data.

11.4.2 Vector Data

Many of the features and tools available in QGIS work the same, regardless of the vector data source. However, because of the differences in format specifications (GeoPackage, ESRI Shapefile, MapInfo and MicroStation file formats, AutoCAD DXF, PostgreSQL, SpatiaLite, Oracle Spatial, MS SQL Server, SAP HANA Spatial databases and many more), QGIS may handle some of their properties differently. Support is provided by the GDAL vector drivers. This section describes how to work with some of these specifics.

> **Pastaba**
>
> QGIS supports (multi)point, (multi)line, (multi)polygon, CircularString, CompoundCurve, CurvePolygon, MultiCurve, MultiSurface feature types, all optionally with Z and/or M values.
>
> You should also note that some drivers don't support some of these feature types, like CircularString, CompoundCurve, CurvePolygon, MultiCurve, MultiSurface feature type. QGIS will convert them.

GeoPackage

The GeoPackage (GPKG) format is platform-independent, and is implemented as a SQLite database container, and can be used to store both vector and raster data. The format was defined by the Open Geospatial Consortium (OGC), and was published in 2014.

GeoPackage can be used to store the following in a SQLite database:

- **vector** features
- **tile matrix sets of imagery** and **raster** maps
- attributes (non-spatial data)
- extensions

Since QGIS version 3.8, GeoPackage can also store QGIS projects. GeoPackage layers can have JSON fields.

GeoPackage is the default format for vector data in QGIS.

ESRI Shapefile format

The ESRI Shapefile format is still one of the most used vector file formats, even if it has some limitations compared to for instance GeoPackage and SpatiaLite.

An ESRI Shapefile format dataset consists of several files. The following three are required:

1. `.shp` file containing the feature geometries
2. `.dbf` file containing the attributes in dBase format
3. `.shx` index file

An ESRI Shapefile format dataset can also include a file with a `.prj` suffix, which contains projection information. While it is very useful to have a projection file, it is not mandatory. A Shapefile format dataset can contain additional files. For further details, see the the ESRI technical specification.

GDAL has read-write support for compressed ESRI Shapefile format (`shz` and `shp.zip`).

Improving Performance for ESRI Shapefile format datasets

To improve the drawing performance for an ESRI Shapefile format dataset, you can create a spatial index. A spatial index will improve the speed of both zooming and panning. Spatial indexes used by QGIS have a `.qix` extension.

Use these steps to create the index:

1. Load an ESRI Shapefile format dataset (see *Naršyklės skydelis*)
2. Open the *Layer Properties* dialog by double-clicking on the layer name in the legend or by right-clicking and choosing *Properties...* from the context menu
3. In the *Source* tab, click the *Create Spatial Index* button

Problem loading a .prj file

If you load an ESRI Shapefile format dataset with a `.prj` file and QGIS is not able to read the coordinate reference system from that file, you will need to define the proper projection manually in the *Layer Properties* ► *Source* tab of the layer by clicking the Select CRS button. This is due to the fact that `.prj` files often do not provide the complete projection parameters as used in QGIS and listed in the *CRS* dialog.

For the same reason, if you create a new ESRI Shapefile format dataset with QGIS, two different projection files are created: a `.prj` file with limited projection parameters, compatible with ESRI software, and a `.qpj` file, providing all the parameters of the CRS. Whenever QGIS finds a `.qpj` file, it will be used instead of the `.prj`.

Delimited Text Files

Delimited text files are very common and widely used because of their simplicity and readability – data can be viewed and edited in a plain text editor. A delimited text file is tabular data with columns separated by a defined character and rows separated by line breaks. The first row usually contains the column names. A common type of delimited text file is a CSV (Comma Separated Values), with columns separated by commas. Delimited text files can also contain positional information (see *Storing geometry information in delimited text files*).

QGIS allows you to load a delimited text file as a layer or an ordinary table (see *Naršyklės skydelis* or *Atskirto tekstinio failo importavimas*). First check that the file meets the following requirements:

1. The file must have a delimited header row of field names. This must be the first line of the data (ideally the first row in the text file).
2. If geometry should be enabled, the file must contain field(s) that define the geometry. These field(s) can have any name.
3. The X and Y coordinates fields (if geometry is defined by coordinates) must be specified as numbers. The coordinate system is not important.
4. If you have a CSV file with non-string columns, you can have an accompanying CSVT file (see section *Using CSVT file to control field formatting*).

The elevation point data file `elevp.csv` in the QGIS sample dataset (see section *Pavyzdinių duomenų atsisiuntimas*) is an example of a valid text file:

```
X;Y;ELEV
-300120;7689960;13
-654360;7562040;52
1640;7512840;3
[...]
```

Some things to note about the text file:

1. The example text file uses `;` (semicolon) as delimiter (any character can be used to delimit the fields).
2. The first row is the header row. It contains the fields `X`, `Y` and `ELEV`.
3. No quotes (`"`) are used to delimit text fields
4. The X coordinates are contained in the `X` field
5. The Y coordinates are contained in the `Y` field

Storing geometry information in delimited text files

Delimited text files can contain geometry information in two main forms:

- As coordinates in separate columns (eg. `Xcol`, `Ycol`...), for point geometry data;
- As well-known text (WKT) representation of geometry in a single column, for any geometry type.

Features with curved geometries (CircularString, CurvePolygon and CompoundCurve) are supported. Here are some examples of geometry types in a delimited text file with geometries coded as WKT:

```
Label;WKT_geom
LineString;LINESTRING(10.0 20.0, 11.0 21.0, 13.0 25.5)
CircularString;CIRCULARSTRING(268 415,227 505,227 406)
CurvePolygon;CURVEPOLYGON(CIRCULARSTRING(1 3, 3 5, 4 7, 7 3, 1 3))
CompoundCurve;COMPOUNDCURVE((5 3, 5 13), CIRCULARSTRING(5 13, 7 15,
  9 13), (9 13, 9 3), CIRCULARSTRING(9 3, 7 1, 5 3))
```

Delimited text files also support Z and M coordinates in geometries:

```
LINESTRINGZ(10.0 20.0 30.0, 11.0 21.0 31.0, 11.0 22.0 30.0)
```

Using CSVT file to control field formatting

When loading CSV files, the GDAL driver assumes all fields are strings (i.e. text) unless it is told otherwise. You can create a CSVT file to tell GDAL (and QGIS) the data type of the different columns:

Tipas	Pavadinimas	Example
Whole number	Sveikas skaičius	4
Boolean	Integer(Boolean)	true
Decimal number	Real	3.456
Data	Date (YYYY-MM-DD)	2016-07-28
Time	Time (HH:MM:SS+nn)	18:33:12+00
Date & Time	DateTime (YYYY-MM-DD HH:MM:SS+nn)	2016-07-28 18:33:12+00
CoordX	CoordX	8.8249
CoordY	CoordY	47.2274
Point(X)	Point(X)	8.8249
Point(Y)	Point(Y)	47.2274
WKT	WKT	POINT(15 20)

The CSVT file is a **ONE line** plain text file with the data types in quotes and separated by commas, e.g.:

```
"Integer","Real","String"
```

You can even specify width and precision of each column, e.g.:

```
"Integer(6)","Real(5.5)","String(22)"
```

This file is saved in the same folder as the `.csv` file, with the same name, but `.csvt` as the extension.

You can find more information at GDAL CSV Driver.

Patarimas

Detect Field Types

Instead of using a CSVT file to tell the data types, QGIS provides the possibility to automatically detect the field types and to change the assumed field types.

PostgreSQL Database

In order to process geographical information in a PostgreSQL database, the PostGIS extension must first be installed. PostGIS extends the capabilities of the PostgreSQL relational database by adding support for storing, indexing, and querying geospatial data.

To enable PostGIS in your database you have to activate the extension in your database (open database and run `CREATE EXTENSION postgis;`). PostGIS enabled databases are also often named „PostGIS layer“. Using PostGIS, vector functions such as select and identify work more accurately than they do with GDAL layers in QGIS.

Patarimas

PostGIS Layers

Normally, a PostGIS layer is identified by an entry in the geometry_columns table. QGIS can load layers that do not have an entry in the geometry_columns table. This includes both tables and views. Refer to your PostgreSQL manual for information on creating views.

This section contains some details on how QGIS accesses PostgreSQL layers. Most of the time, QGIS should simply provide you with a list of database tables that can be loaded, and it will load them on request. However, if you have trouble loading a PostgreSQL table into QGIS, the information below may help you understand QGIS messages and give you directions for modifying the PostgreSQL table or view definition to allow QGIS to load it.

Pastaba

A PostgreSQL database can also store QGIS projects.

Primary key

QGIS requires that PostgreSQL layers contain a column that can be used as a unique key for the layer. For tables, this usually means that the table needs a primary key, or a column with a unique constraint on it. In QGIS, this column needs to be of type int4 (an integer of size 4 bytes). Alternatively, the ctid column can be used as primary key. If a table lacks these items, the oid column will be used instead. Performance will be improved if the column is indexed (note that primary keys are automatically indexed in PostgreSQL).

QGIS offers a checkbox **Select at id** that is activated by default. This option gets the ids without the attributes, which is faster in most cases.

Vaizdas

If the PostgreSQL layer is a view, the same requirement exists, but views do not always have primary keys or columns with unique constraints on them. You have to define a primary key field (has to be integer) in the QGIS dialog before you can load the view. If a suitable column does not exist in the view, QGIS will not load the layer. If this occurs, the solution is to alter the view so that it does include a suitable column (a type of integer and either a primary key or with a unique constraint, preferably indexed).

As for table, a checkbox **Select at id** is activated by default (see above for the meaning of the checkbox). It can make sense to disable this option when you use expensive views.

Pastaba

PostgreSQL foreign table

PostgreSQL foreign tables are not explicitely supported by the PostgreSQL provider and will be handled like a view.

QGIS layer_style table and database backup

If you want to make a backup of your PostgreSQL database using the `pg_dump` and `pg_restore` commands, and the default layer styles as saved by QGIS fail to restore afterwards, you need to set the XML option to `DOCUMENT` before the restore command:

1. Make a PLAIN backup of the `layer_style` table
2. Open the file within a text editor
3. Change the line `SET xmloption = content;` into `SET XML OPTION DOCUMENT;`
4. Save the file
5. Use psql to restore the table in the new database

Filter database side

QGIS allows to filter features already on server side. Check *Settings* ► *Options* ► *Data Sources* ► *Execute expressions on server-side if possible* to do so. Only supported expressions will be sent to the database. Expressions using unsupported operators or functions will gracefully fallback to local evaluation.

Support of PostgreSQL data types

Data types supported by the PostgreSQL provider include: integer, float, boolean, binary object, varchar, geometry, timestamp, array, hstore and json.

Importing Data into PostgreSQL

Data can be imported into PostgreSQL using several tools, including the Browser, DB Manager plugin and the command line tools shp2pgsql and ogr2ogr.

DB tvarkyklė

QGIS comes with a core plugin named DB Manager. It can be used to load data, and it includes support for schemas. See section *DB Manager Plugin* for more information.

shp2pgsql

PostGIS includes a utility called **shp2pgsql**, that can be used to import Shapefile format datasets into a PostGIS-enabled database. For example, to import a Shapefile format dataset named `lakes.shp` into a PostgreSQL database named `gis_data`, use the following command:

```
shp2pgsql -s 2964 lakes.shp lakes_new | psql gis_data
```

This creates a new layer named `lakes_new` in the `gis_data` database. The new layer will have a spatial reference identifier (SRID) of 2964. See section *Darbas su projekcijomis* for more information about spatial reference systems and projections.

Patarimas

Exporting datasets from PostgreSQL

There is also a tool for exporting postgreSQL datasets wth geographic data to Shapefile format: **pgsql2shp**. It is shipped within your PostGIS distribution.

ogr2ogr

In addition to **shp2pgsql** and **DB Manager**, there is another tool for feeding geographical data in PostgreSQL: **ogr2ogr**. It is part of your GDAL installation.

To import a Shapefile format dataset into PostgreSQL, do the following:

```
ogr2ogr -f "PostgreSQL" PG:"dbname=postgis host=myhost.de user=postgres
password=topsecret" alaska.shp
```

This will import the Shapefile format dataset `alaska.shp` into the database *postgis* using the user *postgres* with the password *topsecret* on the host server *myhost.de*.

Note that GDAL must be built with PostgreSQL. You can verify this by typing (in):

```
ogrinfo --formats | grep -i post
```

If you prefer to use the PostgreSQL's **COPY** command instead of the default **INSERT INTO** method, you can export the following environment variable (at least available on and):

```
export PG_USE_COPY=YES
```

ogr2ogr does not create spatial indexes like **shp2pgsl** does. You need to create them manually, using the normal SQL command **CREATE INDEX** afterwards, as an extra step (as described in the next section *Improving Performance*).

Improving Performance

Retrieving features from a PostgreSQL database can be time-consuming, especially over a network. You can improve the drawing performance of PostgreSQL layers by ensuring that a spatial index exists on each layer in the database. PostGIS supports creation of a GiST (Generalized Search Tree) index to speed up spatial searching (GiST index information is taken from the PostGIS documentation available at https://postgis.net).

Patarimas

You can use the DBManager to create an index for your layer. You should first select the layer and click on *Table* ► *Edit table*, go to *Indexes* tab and click on *Add Spatial Index*.

The syntax for creating a GiST index is:

```
CREATE INDEX [indexname] ON [tablename]
  USING GIST ( [geometryfield] GIST_GEOMETRY_OPS );
```

Note that for large tables, creating the index can take a long time. Once the index is created, you should perform a `VACUUM ANALYZE`. See the PostGIS documentation (POSTGIS-PROJECT in *Literature and Web References*) for more information.

The following example creates a GiST index:

```
gsherman@madison:~/current$ psql gis_data
Welcome to psql 8.3.0, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

gis_data=# CREATE INDEX sidx_alaska_lakes ON alaska_lakes
gis_data-# USING GIST (geom GIST_GEOMETRY_OPS);
CREATE INDEX
gis_data=# VACUUM ANALYZE alaska_lakes;
VACUUM
gis_data=# \q
gsherman@madison:~/current$
```

SpatiaLite sluoksniai

If you want to save a vector layer using the SpatiaLite format, you can do this by following instructions at *Creating new layers from an existing layer*. You select `SpatiaLite` as *Format* and enter both *File name* and *Layer name*.

Also, you can select `SQLite` as format and then add `SPATIALITE=YES` in the *Custom Options* ► *Data source* field. This tells GDAL to create a SpatiaLite database. See also https://gdal.org/en/latest/drivers/vector/sqlite.html.

QGIS also supports editable views in SpatiaLite. For SpatiaLite data management, you can also use the core plugin *DB Manager*.

If you want to create a new SpatiaLite layer, please refer to section *Naujo SpatiaLite sluoksnio kūrimas*.

GeoJSON specific parameters

When *exporting layers* to GeoJSON, there are some specific *Layer Options* available. These options come from GDAL which is responsible for the writing of the file:

- *COORDINATE_PRECISION* the maximum number of digits after the decimal separator to write in coordinates. Defaults to 15 (note: for Lat Lon coordinates 6 is considered enough). Truncation will occur to remove trailing zeros.
- *RFC7946* by default GeoJSON 2008 will be used. If set to YES, the updated RFC 7946 standard will be used. Default is NO (thus GeoJSON 2008). See https://gdal.org/en/latest/drivers/vector/geojson.html#rfc-7946-write-support for the main differences, in short: only EPSG:4326 is allowed, other crs's will be transformed, polygons will be written such as to follow the right-hand rule for orientation, values of a „bbox" array are [west, south, east, north], not [minx, miny, maxx, maxy]. Some extension member names are forbidden in FeatureCollection, Feature and Geometry objects, the default coordinate precision is 7 decimal digits
- *WRITE_BBOX* set to YES to include the bounding box of the geometries at the feature and feature collection level

Besides GeoJSON there is also an option to export to „GeoJSON - Newline Delimited" (see https://gdal.org/en/latest/drivers/vector/geojsonseq.html). Instead of a FeatureCollection with Features, you can stream one type (probably only Features) sequentially separated with newlines.

GeoJSON - Newline Delimited has some specific Layer options availabe too:

- *COORDINATE_PRECISION* see above (same as for GeoJSON)
- *RS* whether to start records with the RS=0x1E character. The difference is how the features are separated: only by a newline (LF) character (Newline Delimited JSON, geojsonl) or by also prepending a record-separator (RS) character (giving GeoJSON Text Sequences, geojsons). Default to NO. Files are given the `.json` extension if extension is not provided.

SAP HANA Spatial Layers

This section contains some details on how QGIS accesses SAP HANA layers. Most of the time, QGIS should simply provide you with a list of database tables and views that can be loaded, and it will load them on request. However, if you have trouble loading an SAP HANA table or view into QGIS, the information below may help you understand the root cause and assist in resolving the issue.

Feature Identification

If you'd like to use all of QGIS' feature editing capabilities, QGIS must be able to unambiguously identify each feature in a layer. Internally, QGIS uses a 64-bit signed integer to identify features, whereas the negative range is reserved for special purposes.

Therefore, the SAP HANA provider requires a unique key that can be mapped to a positive 64-bit integer to fully support QGIS' feature editing capabilities. If it is not possible to create such a mapping, you might still view the features, but editing might not work.

Adding tables

When adding a table as a layer, the SAP HANA provider uses the table's primary key to map it to a unique feature id. Therefore, to have full feature editing support, you need to have a primary key to your table definition.

The SAP HANA provider supports multi-column primary keys, but if you'd like to get the best performance, your primary key should be a single column of type `INTEGER`.

Adding views

When adding a view as a layer, the SAP HANA provider cannot automatically identify columns that unambiguously identify a feature. Furthermore, some views are read-only and cannot be edited.

To have full feature editing support, the view must be updatable (check column `IS_READ_ONLY` in system view `SYS.VIEWS` for the view in question) and you must manually provide QGIS with one or more columns that identify a feature. The columns can be given by using *Layer* ► *Add Layer* ► *Add SAP HANA Spatial Layer* and then selecting the columns in the *Feature id* column. For best performance, the *Feature id* value should be a single `INTEGER` column.

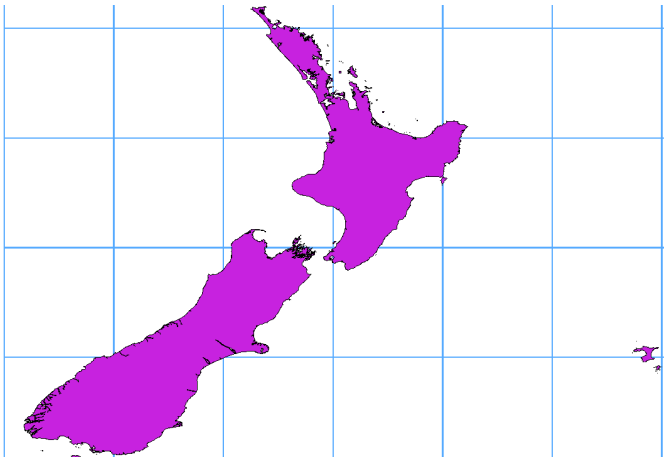
11.4.3 Layers crossing 180° longitude

Many GIS packages don't wrap layers with a geographic reference system (lat/lon) crossing the 180 degrees longitude line. As result, if we open such a layer in QGIS, we could see two widely separated locations, that should appear near each other. In Fig. 11.42, the tiny point on the far left of the map canvas (Chatham Islands) should be within the grid, to the right of the New Zealand main islands.

11.42 Fig. : Map in lat/lon crossing the 180° longitude line

Solving in PostgreSQL

A work-around is to transform the longitude values using PostgreSQL and the ST_ShiftLongitude function. This function reads every point/vertex in every component of every feature in a geometry, and shifts its longitude coordinate from -180..0° to 180..360° and vice versa if between these ranges. This function is symmetrical so the result is a 0..360° representation of a -180..180° data and a -180..180° representation of a 0..360° data.



11.43 Fig. : Crossing 180° longitude applying the **ST_ShiftLongitude** function

1. Import data into PostgreSQL (*Importing Data into PostgreSQL*) using, for example, the DB Manager plugin.
2. Use the PostgreSQL command line interface to issue the following command:

```
-- In this example, "TABLE" is the actual name of your PostgreSQL table
update TABLE set geom=ST_ShiftLongitude(geom);
```

3. If everything went well, you should receive a confirmation about the number of features that were updated. Then you'll be able to load the map and see the difference (*Figure_vector_crossing_map*).

SKYRIUS 12

Working with Vector Data

12.1 Vektorių savybių dialogas

Vektorinių sluoksnių *Sluoksnio savybių* dialogas teikia bendrus nustatymus, leidžiančius valdyti sluoksnio geoobjektų išvaizdą (simbologiją, užrašus, diagramas) žemėlapyje, sąveiką su pele (veiksmus, žemėlapio patarimus, formos išvaizdą). Jis taipogi teikia informaciją apie sluoksnį.

Norėdami pasiekti *Sluoksnio savybių* dialogą:

- Skydelyje *Sluoksniai* du kartus spauskite sluoksnį arba spauskite dešiniu pelės mygtuku ir tada iššokančiame meniu parinkite *Savybės...*;
- Parinkus sluoksnį eikite į meniu *Sluoksniai* ► *Sluoksnio savybės...*.

Vektorinio *Sluoksnio savybių* dialoge yra šios skiltys:

Informacija	*Šaltinis*	*Symbology* (1)
Labels (1)	*Mask* (1)	*3D View* (1)
Diagramos	*Laukai*	*Atributų forma*
Sujungimai	*Pagalbinė saugykla*	*Veiksmai*
Rodymas	*Braižymas*	*Laikas*
Kintamieji	*Aukštis*	*Metaduomenys*
Priklausomybės	*Sutartiniai ženklai*	*QGIS Serveris*
Skaitmeninimas	*External plugins* (2) tabs	

[1] Also available in the *Layer styling panel*
[2] *External plugins* you install can optionally add tabs to this dialog. Those are not presented in this document. Refer to their documentation.

Patarimas

Bendrinkite visas ar dalį sluoksnio stilių savybių

Dialogo apačioje esantis meniu *Stilius* leidžia jums importuoti ir eksportuoti šias ar dalį šių savybių savybių iš/į keletą paskirčių (failą, iškarpinę, duomenų bazę). Daugiau informacijos rasite skyriuje *Savo stilių tvarkymas*.

Pastaba

Kadangi įtrauktų sluoksnių (žr. *Išorinių projektų sluoksnių įtraukimas*) savybės (simbologija, užrašai, veiksmai, numatytos reikšmės, formos…) traukiamos iš originalaus projekto failo, kad būtų išvengta pakeitimų, kurie gali sulaužyti šią elgseną, sluoksnio savybių dialogas šiems sluoksniams neprieinamas.

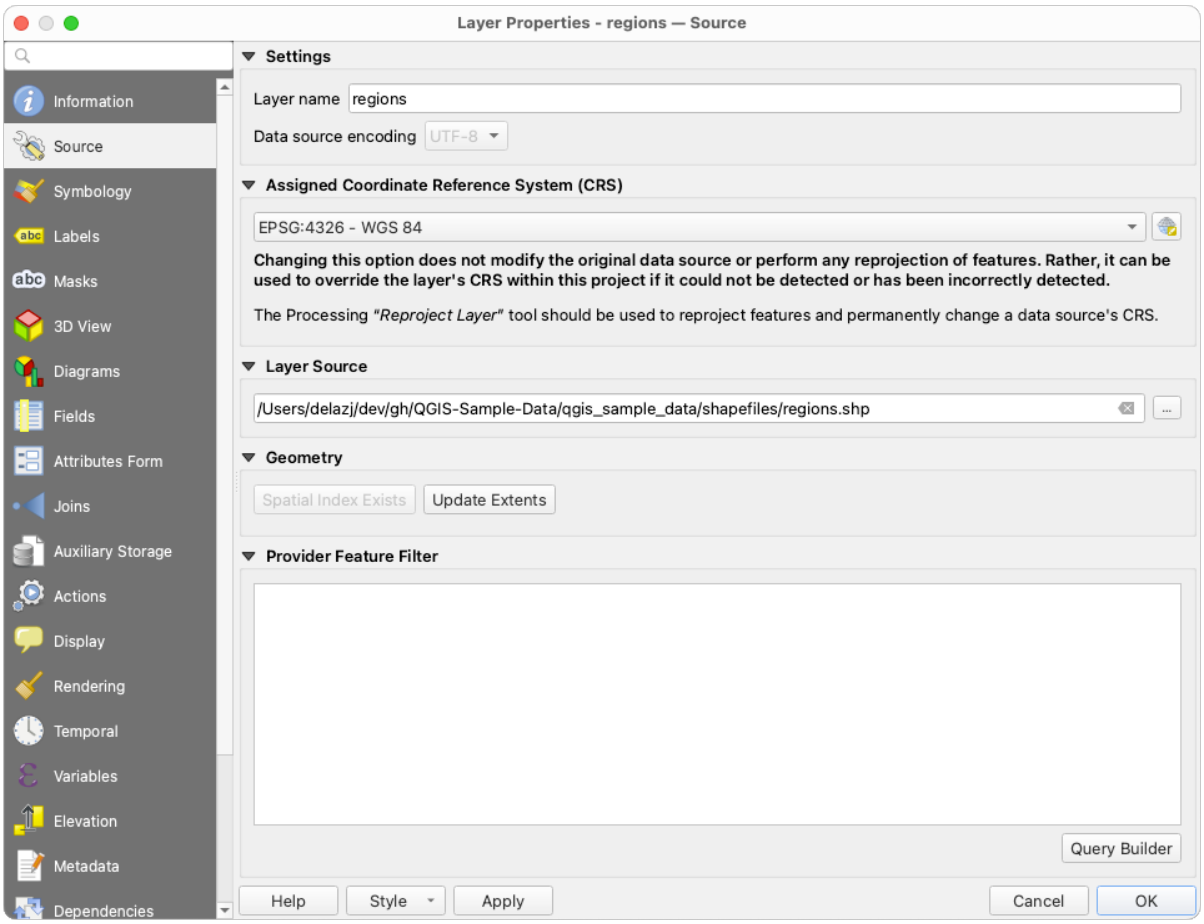
12.1.1 Informacinės savybės

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata on the current layer. Provided information are:

- general such as name in the project, source path, list of auxiliary files, last save time and size, the used provider
- custom properties, used to store in the active project additional information about the layer. Default custom properties may include *layer notes*, *legend widgets*, *layer variables*, *form properties*… More custom properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- based on the provider of the layer: format of storage, geometry type, data source encoding, extent, feature count…
- the Coordinate Reference System: name, units, method, accuracy, reference (i.e. whether it's static or dynamic)
- picked from the *filled metadata*: access, extents, links, contacts, history…
- and related to its geometry (spatial extent, CRS…) or its attributes (number of fields, characteristics of each…).

12.1.2 Source Properties

Use this tab to define general settings for the vector layer.



12.1 Fig. : Source tab in vector Layer Properties dialog

Nustatymai

- Set a *Layer name* different from the layer filename that will be used to identify the layer in the project (in the *Layers Panel*, with expressions, in print layout legend, …)
- Depending on the data format, select the *Data source encoding* if not correctly detected by QGIS.

Coordinate Reference System

- Displays the layer's *Assigned Coordinate Reference System (CRS)*. You can change the layer's CRS, selecting a recently used one in the drop-down list or clicking on Select CRS button (see *Koordinačių atskaitos sistemos parinkiklis*). Use this process only if the CRS applied to the layer is a wrong one or if none was applied. If you wish to reproject your data into another CRS, rather use layer reprojection algorithms from Processing or *Save it into another layer*.

Layer source and Geometry

- Depending on the data provider, a *Layer source* group indicates the path to the source of the dataset and allows for replacing the loaded layer:
  - When the layer is stored as file on disk, edit the path shown in the text box or press … Browse to select another file on the disk. Both layers do not need to share attribute fields, geometry type or file formats.
  - When the layer is provided by an ArcGIS Feature service, it is possible to modify its *authentication settings*, while keeping unchanged the details for connecting to the service.
- *Create spatial index* (only for OGR-supported formats): helps speeding layer rendering and features' geometry retrieval.
- *Update extents* information for a layer.

Query Builder

The Query Builder dialog is accessible through the *Query Builder* button at the bottom of the *Source* tab in the Layer Properties dialog, under the *Provider feature filter* group.

The Query Builder provides an interface that allows you to define a subset of the features in the layer using a SQL-like WHERE clause and to display the result in the main window. As long as the query is active, only the features corresponding to its result are available in the project.

You can use one or more layer attributes to define the filter in the `Query Builder`. The use of more than one attribute is shown in Fig. 12.2. In the example, the filter combines the attributes

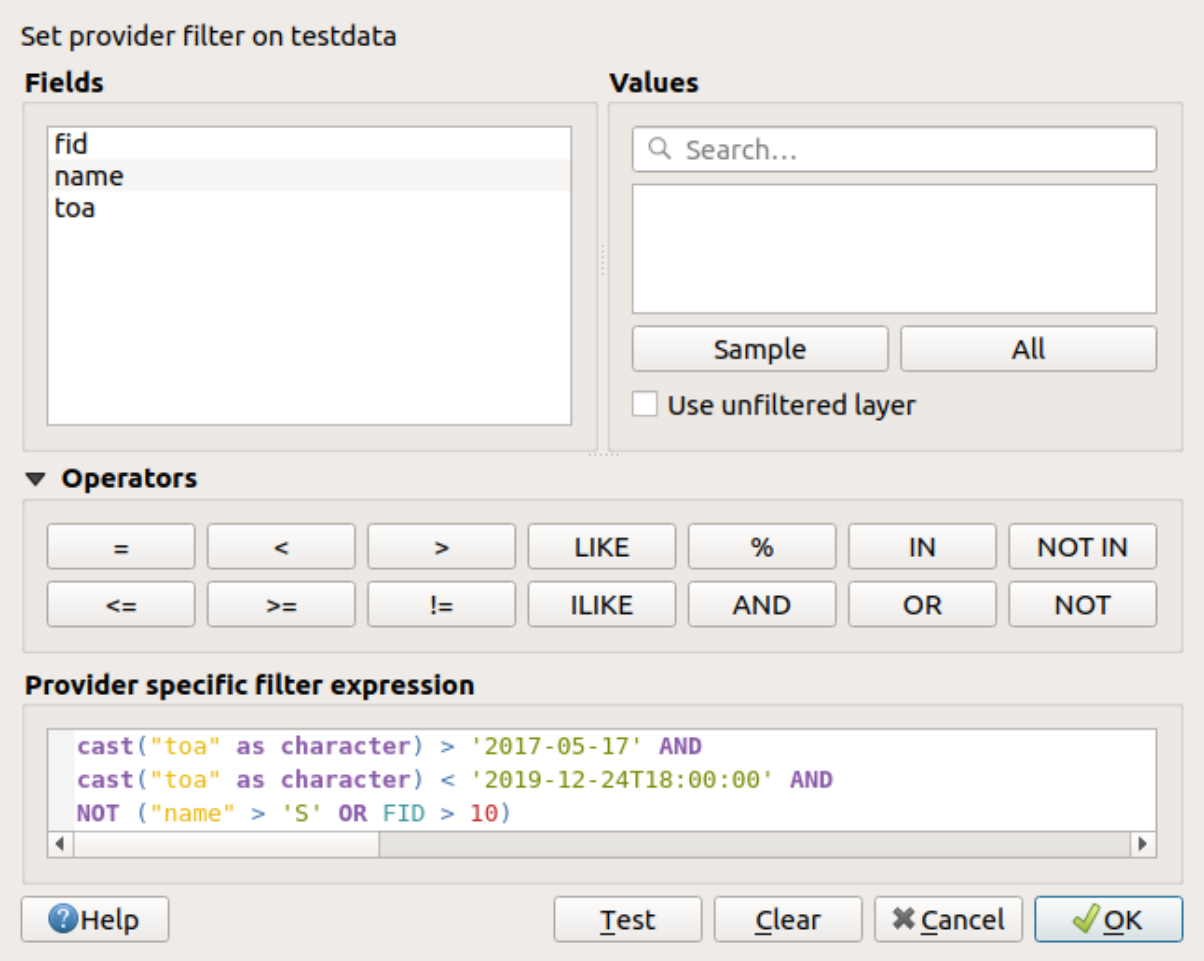
- `toa` (`DateTime` field: `cast("toa" as character) > '2017-05-17'` and `cast("toa" as character) < '2019-12-24T18:00:00'`),
- `name` (`String` field: `"name" > 'S'`) and
- `FID` (`Integer` field: `FID > 10`)

using the AND, OR and NOT operators and parenthesis. This syntax (including the DateTime format for the `toa` field) works for GeoPackage datasets.

The filter is made at the data provider (OGR, PostgreSQL, MS SQL Server…) level. So the syntax depends on the data provider (DateTime is for instance not supported for the ESRI Shapefile format). The complete expression:

```
cast("toa" as character) > '2017-05-17' AND
cast("toa" as character) < '2019-12-24T18:00:00' AND
NOT ("name" > 'S' OR FID > 10)
```

You can also open the *Query Builder* dialog using the *Filter…* option from the *Layer* menu or the layer contextual menu. The *Fields*, *Values* and *Operators* sections in the dialog help you to construct the SQL-like query exposed in the *Provider specific filter expression* box.



12.2 Fig. : Query Builder

The **Fields** list contains all the fields of the layer. To add an attribute column to the SQL WHERE clause field, double-click its name or just type it into the SQL box.

The **Values** frame lists the values of the currently selected field. To list all unique values of a field, click the *All* button. To instead list the first 25 unique values of the column, click the *Sample* button. To add a value to the SQL WHERE clause field, double click its name in the Values list. You can use the search box at the top of the Values frame to easily browse and find attribute values in the list.

The **Operators** section contains all usable operators. To add an operator to the SQL WHERE clause field, click the appropriate button. Relational operators (= , > , …), string comparison operator (`LIKE`), and logical operators (`AND`, `OR`, …) are available.

The *Test* button helps you check your query and displays a message box with the number of features satisfying the current query. Use the *Clear* button to wipe the SQL query and revert the layer to its original state (ie, fully load all the features). It is possible to *Save…* the query as a `.QQF` file, or *Load…* the query from a file into the dialog.

When a filter is applied, QGIS treats the resulting subset acts as if it were the entire layer. For example if you applied the filter above for ‚Borough' (`"TYPE_2" = 'Borough'`), you can not display, query, save or edit `Anchorage`, because that is a ‚Municipality' and therefore not part of the subset.

Patarimas

Filtered layers are indicated in the Layers Panel

In the *Layers* panel, filtered layer is listed with a Filter icon next to it indicating the query used when the mouse

hovers over the button. Double-click the icon opens the *Query Builder* dialog for edit. This can also be achieved through the *Layer* ► *Filter…* menu.

12.1.3 Symbology Properties

The *Symbology* tab provides you with a comprehensive tool for rendering and symbolizing your vector data. You can use tools that are common to all vector data, as well as special symbolizing tools that were designed for the different kinds of vector data. However all types share the following dialog structure: in the upper part, you have a widget that helps you prepare the classification and the symbol to use for features and at the bottom the *Layer rendering* widget.

Patarimas

Switch quickly between different layer representations

Using the *Styles* ► *Add* menu at the bottom of the *Layer Properties* dialog, you can save as many styles as needed. A style is the combination of all properties of a layer (such as symbology, labeling, diagram, fields form, actions…) as you want. Then, simply switch between styles from the context menu of the layer in *Layers Panel* to automatically get different representations of your data.

Patarimas

Export vector symbology

You have the option to export vector symbology from QGIS into Google *.kml, *.dxf and MapInfo *.tab files. Just open the right mouse menu of the layer and click on *Save As…* to specify the name of the output file and its format. In the dialog, use the *Symbology export* menu to save the symbology either as *Feature symbology* ► or as *Symbol layer symbology* ►. If you have used symbol layers, it is recommended to use the second setting.

Features rendering

The renderer is responsible for drawing a feature together with the correct symbol. Regardless layer geometry type, there are four common types of renderers: single symbol, categorized, graduated and rule-based. For point layers, there are point displacement, point cluster and heatmap renderers available while polygon layers can also be rendered with the merged features, inverted polygons and 2.5 D renderers.

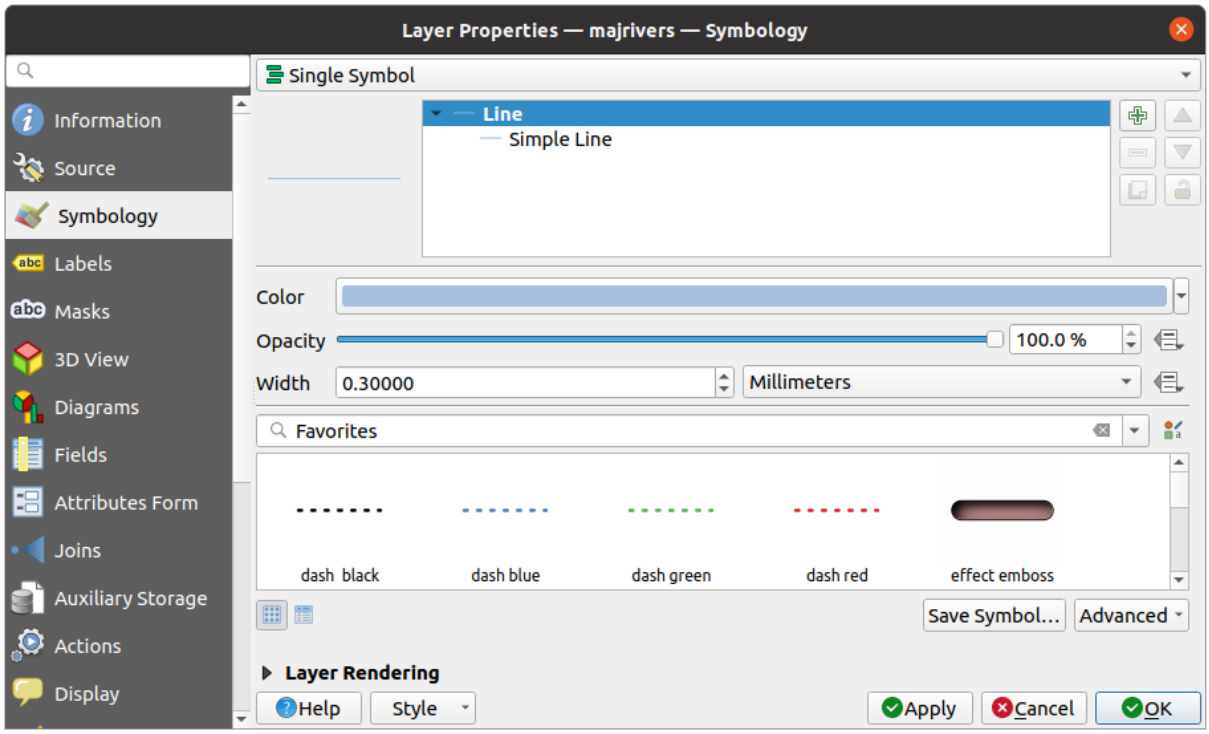
There is no continuous color renderer, because it is in fact only a special case of the graduated renderer. The categorized and graduated renderers can be created by specifying a symbol and a color ramp - they will set the colors for symbols appropriately. For each data type (points, lines and polygons), vector symbol layer types are available. Depending on the chosen renderer, the dialog provides different additional sections.

Pastaba

If you change the renderer type when setting the style of a vector layer the settings you made for the symbol will be maintained. Be aware that this procedure only works for one change. If you repeat changing the renderer type the settings for the symbol will get lost.

Single Symbol Renderer

The *Single Symbol* renderer is used to render all features of the layer using a single user-defined symbol. See *Simbolių parinkiklis* for further information about symbol representation.



12.3 Fig. : Single symbol line properties

No Symbols Renderer

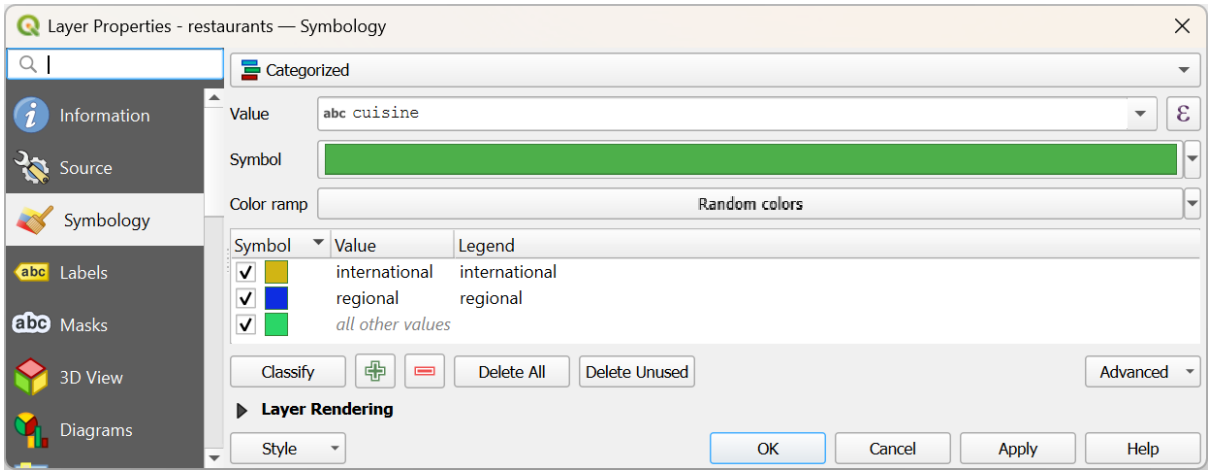
The *No Symbols* renderer is a special use case of the Single Symbol renderer as it applies the same rendering to all features. Using this renderer, no symbol will be drawn for features, but labeling, diagrams and other non-symbol parts will still be shown.

Selections can still be made on the layer in the canvas and selected features will be rendered with a default symbol. Features being edited will also be shown.

This is intended as a handy shortcut for layers which you only want to show labels or diagrams for, and avoids the need to render symbols with totally transparent fill/border to achieve this.

Categorized Renderer

The *Categorized* renderer is used to render the features of a layer, using a user-defined symbol whose aspect reflects the discrete values of a field or an expression.



12.4 Fig. : Categorized Symbolizing options

To use categorized symbology for a layer:

1. Select the *Value* of classification: it can be an existing field or an *expression* you can type in the box or build using the associated ε button. Using expressions for categorizing avoids the need to create a field for symbology purposes only (eg, if your classification criteria are derived from one or more attributes).

   The expression used to classify features can be of any type; eg, it can:

   - be a comparison. In this case, QGIS returns values `1` (**True**) and `0` (**False**). Some examples:

     ```
     myfield >= 100
     $id = @atlas_featureid
     myfield % 2 = 0
     within( $geometry, @atlas_geometry )
     ```

   - combine different fields:

     ```
     concat( field_1, ' ', field_2 )
     ```

   - be a calculation on fields:

     ```
     myfield % 2
     year( myfield )
     field_1 + field_2
     substr( field_1, -3 )
     ```

   - be used to transform linear values to discrete classes, e.g.:

     ```
     CASE WHEN x > 1000 THEN 'Big' ELSE 'Small' END
     ```

   - combine several discrete values into a single category, e.g.:

     ```
     CASE
     WHEN building IN ('residence', 'mobile home') THEN 'residential'
     WHEN building IN ('commercial', 'industrial') THEN 'Commercial and␣
     ↪Industrial'
     END
     ```

Patarimas

While you can use any kind of expression to categorize features, for some complex expressions it might be simpler to use *rule-based rendering*.

2. Configure the *Symbol*, which will be used as base symbol for all the classes;
3. Indicate the *Color ramp*, i.e. the range of colors from which the color applied to each symbol is selected.

   Besides the common options of the *color ramp widget*, you can apply a *Random Color Ramp* to the categories. You can click the *Shuffle Random Colors* entry to regenerate a new set of random colors if you are not satisfied.
4. Then click on the *Classify* button to create classes from the distinct values of the provided field or expression.
5. For each class, you can edit the *Legend* column to a more meaningful label (used in the *Layers* panel and the print layout).

   Pastaba

   Keep in mind that some values may use widgets that do not display the actual value stored in the field. For example, a checkbox widget may store `1` and `0` for checked and unchecked states, while displaying `True` and `False` labels. In this case, to categorize features based on the checkbox state, you need to use the stored values (`1` and `0`) in the expression. QGIS will automatically use the display value for the legend column.
6. *Apply* the changes if the *live update* is not in use and each feature on the map canvas will be rendered with the symbol of its class.

   By default, QGIS appends an *all other values* class to the list. While empty at the beginning, this class is used as a default class for any feature not falling into the other classes (eg, when you create features with new values for the classification field / expression).

Further tweaks can be done to the default classification:

- You can Add new categories, Remove selected categories, *Delete All* of them or *Delete Unused* categories.
- A class can be disabled by unchecking the checkbox to the left of the class name; the corresponding features are hidden on the map.
- Drag-and-drop the rows to reorder the classes
- To change the symbol, the value or the legend of a class, double click the item.

Right-clicking over selected item(s) shows a contextual menu to:

- *Copy Symbol* and *Paste Symbol*, a convenient way to apply the item's representation to others
- *Change Color…* of the selected symbol(s)
- *Change Opacity…* of the selected symbol(s)
- *Change Output Unit…* of the selected symbol(s)
- *Change Width…* of the selected line symbol(s)
- *Change Size…* of the selected point symbol(s)
- *Change Angle…* of the selected point symbol(s)
- *Merge Categories*: Groups multiple selected categories into a single one. This allows simpler styling of layers with a large number of categories, where it may be possible to group numerous distinct categories into a smaller and more manageable set of categories which apply to multiple values.

  Patarimas

  Since the symbol kept for the merged categories is the one of the topmost selected category in the list, you may want to move the category whose symbol you wish to reuse to the top before merging.
- *Unmerge Categories* that were previously merged

The created classes also appear in a tree hierarchy in the *Layers* panel. Double-click an entry in the map legend to edit the assigned symbol. Right-click and you will get some *more options*.

The *Advanced* menu gives access to options to speed classification or fine-tune the symbols rendering:

- *Match to saved symbols*: Using the *symbols library*, assigns to each category a symbol whose name represents the classification value of the category
- *Match to symbols from file…*: Provided a file with symbols, assigns to each category a symbol whose name represents the classification value of the category
- *Symbol levels…* to define the order of symbols rendering.

Graduated Renderer

The *Graduated* renderer is used to render all the features from a layer, using an user-defined symbol whose color or size reflects the assignment of a selected feature's attribute to a class.

Like the Categorized Renderer, the Graduated Renderer allows you to define rotation and size scale from specified columns.

Also, analogous to the Categorized Renderer, it allows you to select:

- The *Value* of classification: it can be an existing field or an *expression* you can type in the box or build using the associated button. Using expressions for graduating avoids the need to create a field for symbology purposes only (eg, if your classification criteria are derived from one or more attributes).
- The symbol (using the Symbol selector dialog)
- The legend format and the precision
- The method to use to change the symbol: color or size
- The colors (using the color Ramp list) if the color method is selected
- The size (using the size domain and its unit)

Then you can use the Histogram tab which shows an interactive histogram of the values from the assigned field or expression. Class breaks can be moved or added using the histogram widget.

> **Pastaba**
>
> You can use Statistical Summary panel to get more information on your vector layer. See *Statistinės suvestinės skydelis*.

Back to the Classes tab, you can specify the number of classes and also the mode for classifying features within the classes (using the Mode list). The available modes are:

- Equal Count (Quantile): each class will have the same number of elements (the idea of a boxplot).
- Equal Interval: each class will have the same size (e.g. with the values from 1 to 16 and four classes, each class will have a size of four).
- Fixed Interval: each class will have a fixed range of values (e.g. with the values from 1 to 16 and an interval size of 4, the classes will be 1-4, 5-8, 9-12 and 13-16).
- Logarithmic scale: suitable for data with a wide range of values. Narrow classes for low values and wide classes for large values (e.g. for decimal numbers with range [0..100] and two classes, the first class will be from 0 to 10 and the second class from 10 to 100).
- Natural Breaks (Jenks): the variance within each class is minimized while the variance between classes is maximized.

- Pretty Breaks: computes a sequence of about n+1 equally spaced nice values which cover the range of the values in x. The values are chosen so that they are 1, 2 or 5 times a power of 10. (based on pretty from the R statistical environment https://www.rdocumentation.org/packages/base/topics/pretty).
- Standard Deviation: classes are built depending on the standard deviation of the values.

The listbox in the center part of the *Symbology* tab lists the classes together with their ranges, labels and symbols that will be rendered.

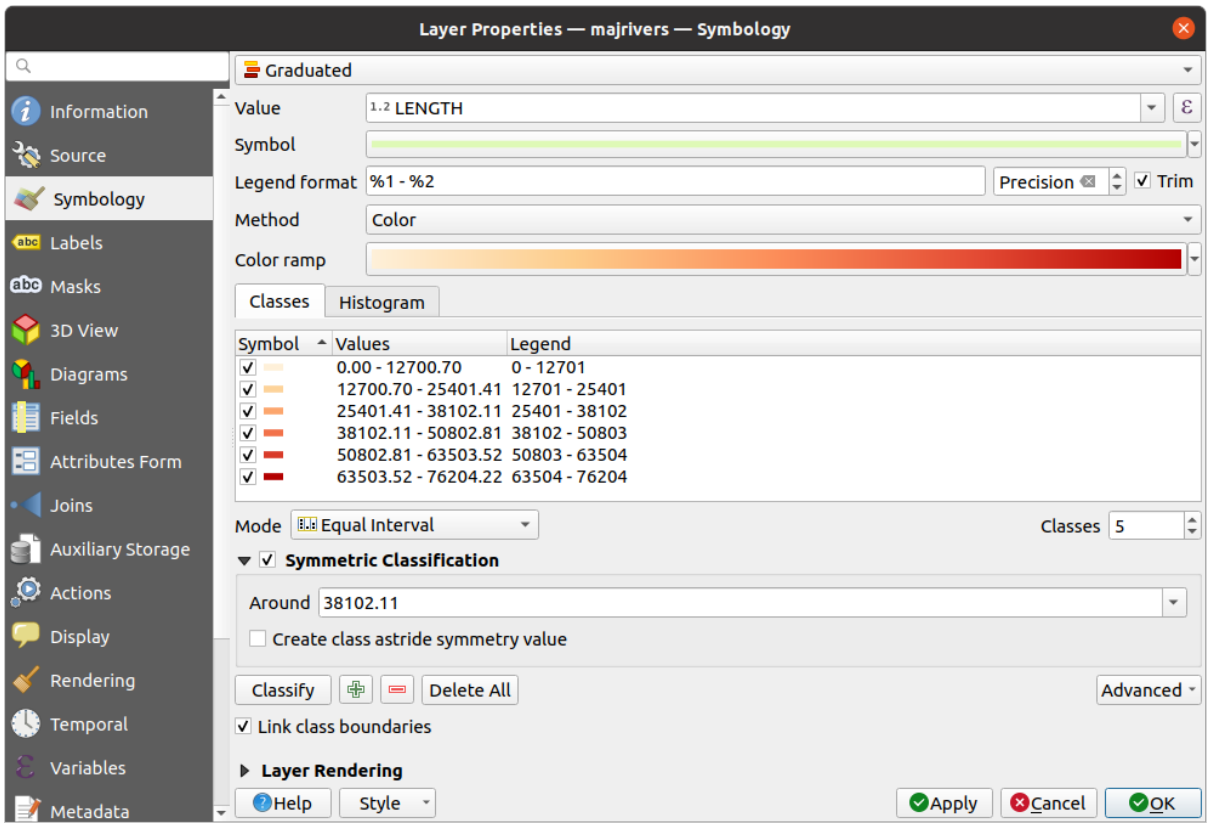
Click on **Classify** button to create classes using the chosen mode. Each classes can be disabled unchecking the checkbox at the left of the class name.

To change symbol, value and/or label of the class, just double click on the item you want to change.

Right-clicking over selected item(s) shows a contextual menu to:

- *Copy Symbol* and *Paste Symbol*, a convenient way to apply the item's representation to others
- *Change Color...* of the selected symbol(s)
- *Change Opacity...* of the selected symbol(s)
- *Change Output Unit...* of the selected symbol(s)
- *Change Width...* of the selected line symbol(s)
- *Change Size...* of the selected point symbol(s)
- *Change Angle...* of the selected point symbol(s)

The example in Fig. 12.5 shows the graduated rendering dialog for the major_rivers layer of the QGIS sample dataset.



12.5 Fig. : Graduated Symbolizing options

The created classes also appear in a tree hierarchy in the *Layers* panel. Double-click an entry in the map legend to edit the assigned symbol. Right-click and you will get some *more options*.

Proportional Symbol and Multivariate Analysis

Proportional Symbol and Multivariate Analysis are not rendering types available from the Symbology rendering drop-down list. However with the *data-defined override* options applied over any of the previous rendering options, QGIS allows you to display your point and line data with such representation.

Creating proportional symbol

To apply a proportional rendering:

1. First apply to the layer the *single symbol renderer*.
2. Then set the symbol to apply to the features.
3. Select the item at the upper level of the symbol tree, and use the Data-defined override *button* next to the *Size* (for point layer) or *Width* (for line layer) option.
4. Select a field or enter an expression, and for each feature, QGIS will apply the output value to the property and proportionally resize the symbol in the map canvas.

   If need be, use the *Size assistant…* option of the menu to apply some transformation (exponential, flannery…) to the symbol size rescaling (see *Nuo duomenų priklausančio permušimo asistento sąsajos naudojimas* for more details).

You can choose to display the proportional symbols in the *Layers panel* and the *print layout legend item*: unfold the *Advanced* drop-down list at the bottom of the main dialog of the *Symbology* tab and select **Data-defined size legend…** to configure the legend items (see *Data-defined size legend* for details).

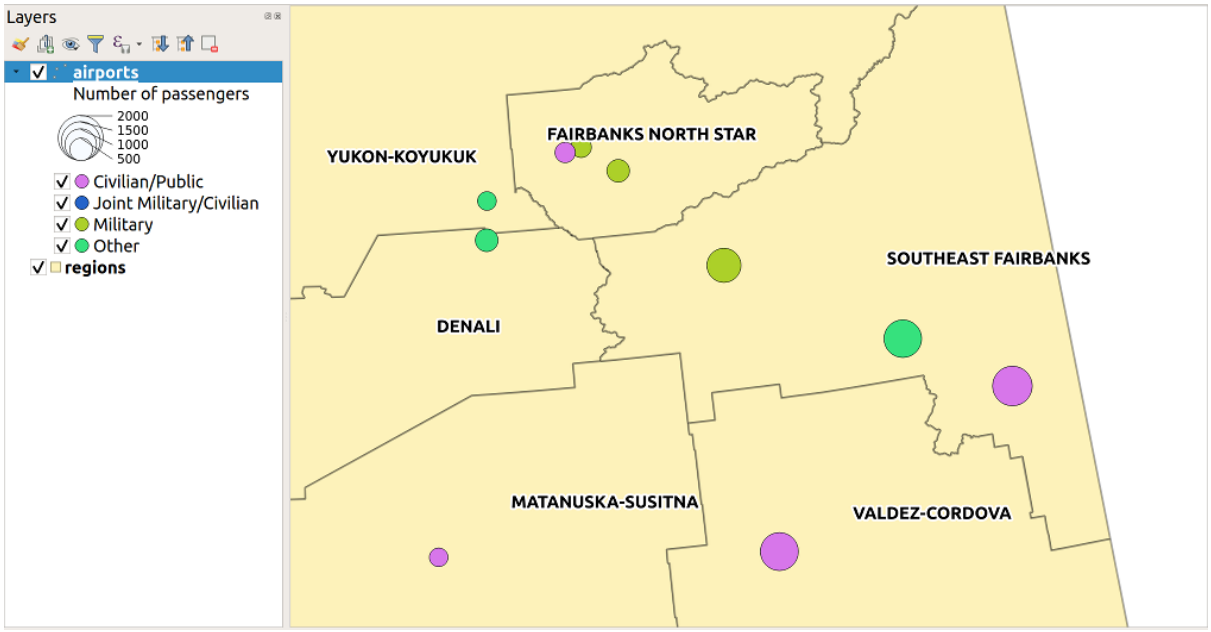
Creating multivariate analysis

A multivariate analysis rendering helps you evaluate the relationship between two or more variables e.g., one can be represented by a color ramp while the other is represented by a size.

The simplest way to create multivariate analysis in QGIS is to:

1. First apply a categorized or graduated rendering on a layer, using the same type of symbol for all the classes.
2. Then, apply a proportional symbology on the classes:
   1. Click on the *Change* button above the classification frame: you get the *Simbolių parinkiklis* dialog.
   2. Rescale the size or width of the symbol layer using the *data defined override* widget as seen above.

Like the proportional symbol, the scaled symbology can be added to the layer tree, on top of the categorized or graduated classes symbols using the *data defined size legend* feature. And both representation are also available in the print layout legend item.



12.6 Fig. : Multivariate example with scaled size legend

Rule-based Renderer

Rules are QGIS *expressions* used to discriminate features according to their attributes or properties in order to apply specific rendering settings to them. Rules can be nested, and features belong to a class if they belong to all the upper nesting level(s).

The *Rule-based* renderer is thus designed to render all the features from a layer, using symbols whose aspect reflects the assignment of a selected feature to a fine-grained class.

To create a rule:

1. Activate an existing row by double-clicking it (by default, QGIS adds a symbol without a rule when the rendering mode is enabled) or click the Edit rule or Add rule button.
2. In the *Edit Rule* dialog that opens, you can define a label to help you identify each rule. This is the label that will be displayed in the *Layers Panel* and also in the print composer legend.
3. Manually enter an expression in the text box next to the *Filter* option or press the button next to it to open the expression string builder dialog.
4. Use the provided functions and the layer attributes to build an *expression* to filter the features you'd like to retrieve. Press the *Test* button to check the result of the query.
5. You can enter a longer label to complete the rule description.
6. You can use the *Scale Range* option to set scales at which the rule should be applied.
7. Finally, configure the *symbol to use* for these features.
8. And press *OK*.

A new row summarizing the rule is added to the Layer Properties dialog. You can create as many rules as necessary following the steps above or copy pasting an existing rule. Drag-and-drop the rules on top of each other to nest them and refine the upper rule features in subclasses.

The rule-based renderer can be combined with categorized or graduated renderers. Selecting a rule, you can organize its features in subclasses using the *Refine selected rules* drop-down menu. Refined classes appear like sub-items of the

rule, in a tree hierarchy and like their parent, you can set the symbology and the rule of each class. Automated rule refinement can be based on:

- **scales**: given a list of scales, this option creates a set of classes to which the different user-defined scale ranges apply. Each new scale-based class can have its own symbology and expression of definition. This can e.g. be a convenient way to display the same features with various symbols at different scales, or display only a set of features depending on the scale (e.g. local airports at large scale vs international airports at small scale).
- **categories**: applies a *categorized renderer* to the features falling in the selected rule.
- or **ranges**: applies a *graduated renderer* to the features falling in the selected rule.

Refined classes appear like sub-items of the rule, in a tree hierarchy and like above, you can set symbology of each class. Symbols of the nested rules are stacked on top of each other so be careful in choosing them. It is also possible to uncheck *Symbols* in the *Edit rule* dialog to avoid rendering a particular symbol in the stack.

In the *Edit rule* dialog, you can avoid writing all the rules and make use of the *Else* option to catch all the features that do not match any of the other rules, at the same level. This can also be achieved by writing `Else` in the *Rule* column of the *Layer Properties* ► *Symbology* ► *Rule-based* dialog.

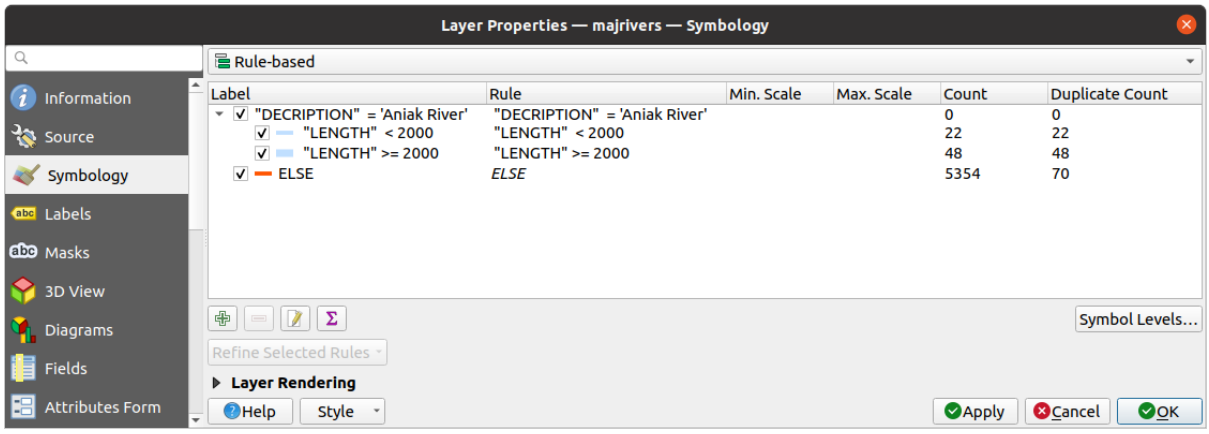
Right-clicking over selected item(s) shows a contextual menu to:

- *Copy* and *Paste*, a convenient way to create new item(s) based on existing item(s)
- *Copy Symbol* and *Paste Symbol*, a convenient way to apply the item's representation to others
- *Change Color…* of the selected symbol(s)
- *Change Opacity…* of the selected symbol(s)
- *Change Output Unit…* of the selected symbol(s)
- *Change Width…* of the selected line symbol(s)
- *Change Size…* of the selected point symbol(s)
- *Change Angle…* of the selected point symbol(s)
- *Refine Current Rule*: open a submenu that allows to refine the current rule with **scales**, **categories** or **Ranges**. Same as selecting the *corresponding menu* at the bottom of the dialog.

Unchecking a row in the rule-based renderer dialog hides in the map canvas the features of the specific rule and the nested ones.

The created rules also appear in a tree hierarchy in the map legend. Double-click an entry in the map legend to edit the assigned symbol. Right-click and you will get some *more options*.

The example in Fig. 12.7 shows the rule-based rendering dialog for the rivers layer of the QGIS sample dataset.



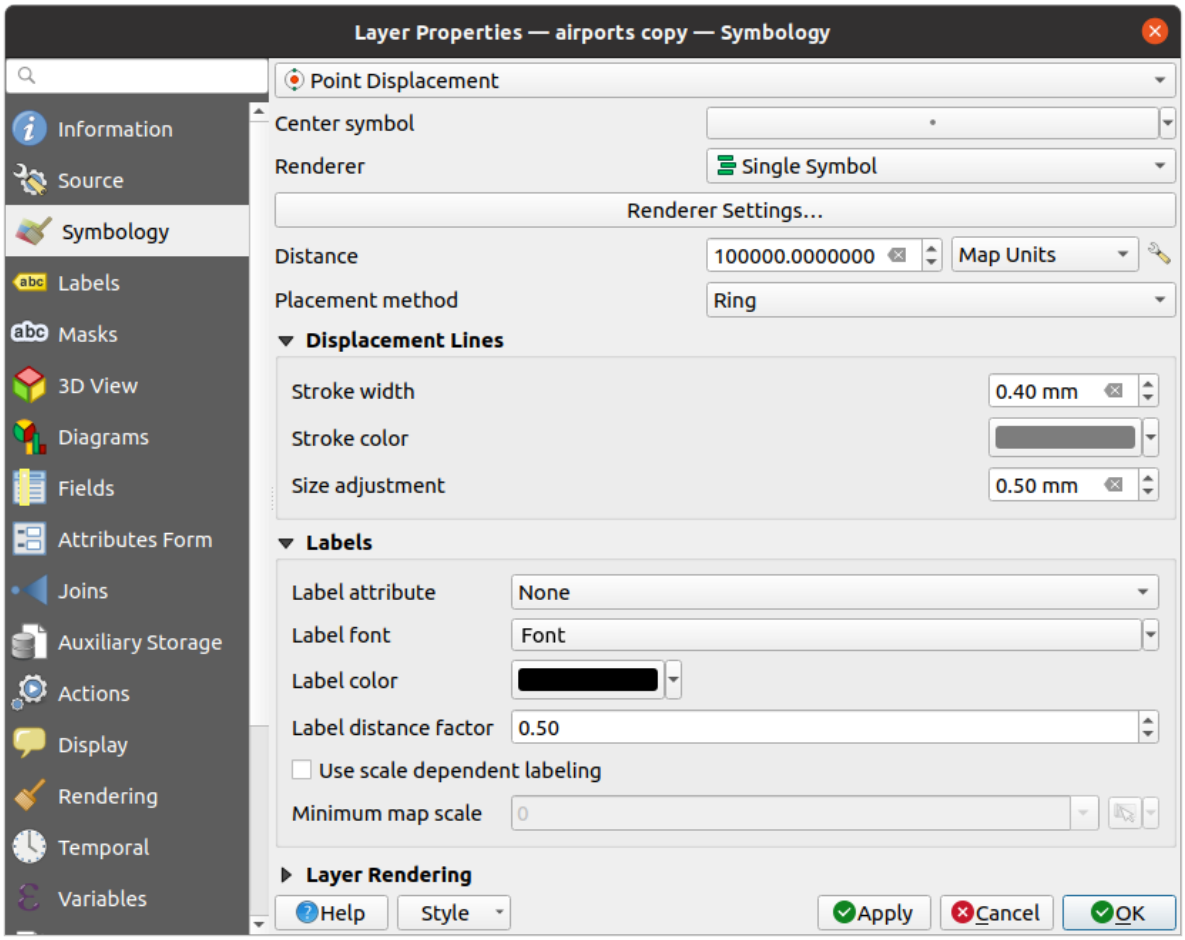
12.7 Fig. : Rule-based Symbolizing options

Point displacement Renderer

The *Point Displacement* renderer takes the point features falling in a given distance tolerance from each other and places their symbols around their barycenter, following different placement methods. This can be a convenient way to visualize all the features of a point layer, even if they have the same location (e.g. amenities in a building).

To configure a point displacement renderer, you have to:

1. Set the *Center symbol*: how the virtual point at the center will look like
2. Select the *Renderer* type: how you want to classify features in the layer (single, categorized, rule-based…)
3. Press the *Renderer Settings…* button to configure features' symbology according to the selected renderer
4. Indicate the *Distance* tolerance in which close features are considered overlapping and then displaced over the same virtual point. Supports common symbol units.
5. Configure the *Placement methods*:
   - **Ring**: places all the features on a circle whose radius depends on the number of features to display.
   - **Concentric rings**: uses a set of concentric circles to show the features.
   - **Grid**: generates a regular grid with a point symbol at each intersection.
6. Displaced symbols are placed on the *Displacement lines*. While the minimal spacing of the displacement lines depends on the point symbols renderer, you can still customize some of their settings such as the *Stroke width*, *Stroke color* and *Size adjustment* (e.g., to add more spacing between the rendered points).
7. Use the *Labels* group options to perform points labeling: the labels are placed near the displaced symbol, and not at the feature real position.
   1. Select the *Label attribute*: a field of the layer to use for labeling
   2. Indicate the *Label font* properties and size
   3. Pick a *Label color*
   4. Set a *Label distance factor*: for each point feature, offsets the label from the symbol center proportionally to the symbol's diagonal size.
   5. Turn on *Use scale dependent labeling* if you want to display labels only on scales larger than a given *Minimum map scale*.



12.8 Fig. : Point displacement dialog

Pastaba

Point Displacement renderer does not alter feature geometry, meaning that points are not moved from their position. They are still located at their initial place. Changes are only visual, for rendering purpose. Use instead the Processing *Points displacement* algorithm if you want to create displaced features.

Point Cluster Renderer

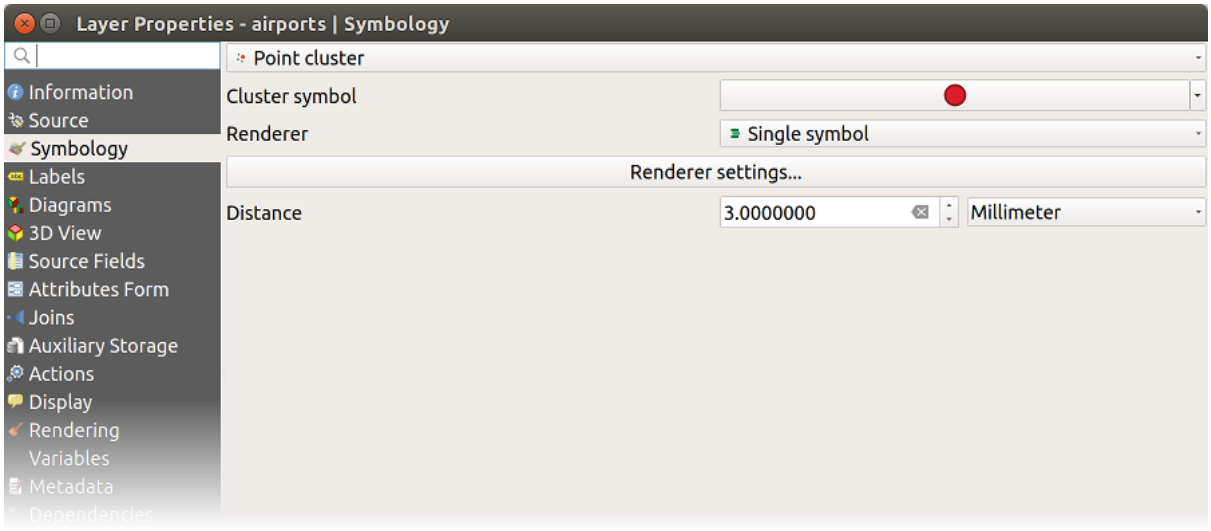
Unlike the *Point Displacement* renderer which blows up nearest or overlaid point features placement, the *Point Cluster* renderer groups nearby points into a single rendered marker symbol. Points that fall within a specified distance from each others are merged into a single symbol. Points aggregation is made based on the closest group being formed, rather than just assigning them the first group within the search distance.

From the main dialog, you can:

1. Set the symbol to represent the point cluster in the *Cluster symbol*; the default rendering displays the number of aggregated features thanks to the `@cluster_size` *variable* on Font marker symbol layer.
2. Select the *Renderer* type, i.e. how you want to classify features in the layer (single, categorized, rule-based…)
3. Press the *Renderer Settings…* button to configure features' symbology as usual. Note that this symbology is only visible on features that are not clustered, the *Cluster symbol* being applied otherwise. Also, when all the point

features in a cluster belong to the same rendering class, and thus would be applied the same color, that color represents the `@cluster_color` variable of the cluster.

4. Indicate the maximal *Distance* to consider for clustering features. Supports common symbol units.



12.9 Fig. : Point Cluster dialog

Pastaba

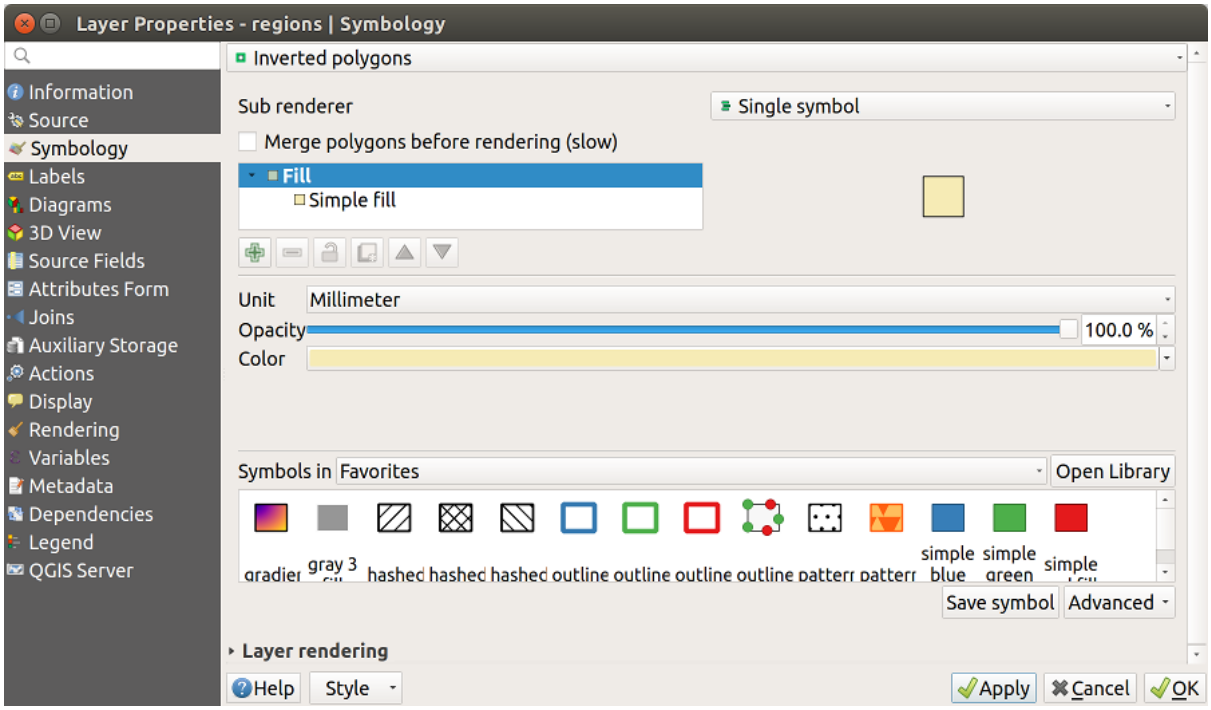
Point Cluster renderer does not alter feature geometry, meaning that points are not moved from their position. They are still located at their initial place. Changes are only visual, for rendering purpose. Use instead the Processing *K-means clustering* or *DBSCAN clustering* algorithm if you want to create cluster-based features.

Merged Features Renderer

The *Merged Features* renderer allows area and line features to be “dissolved” into a single object prior to rendering to ensure that complex symbols or overlapping features are represented by a uniform and contiguous cartographic symbol.

Inverted Polygon Renderer

The *Inverted Polygon* renderer allows user to define a symbol to fill in outside of the layer's polygons. As above you can select subrenderers, namely Single symbol, Graduated, Categorized, Rule-Based or 2.5D renderer.



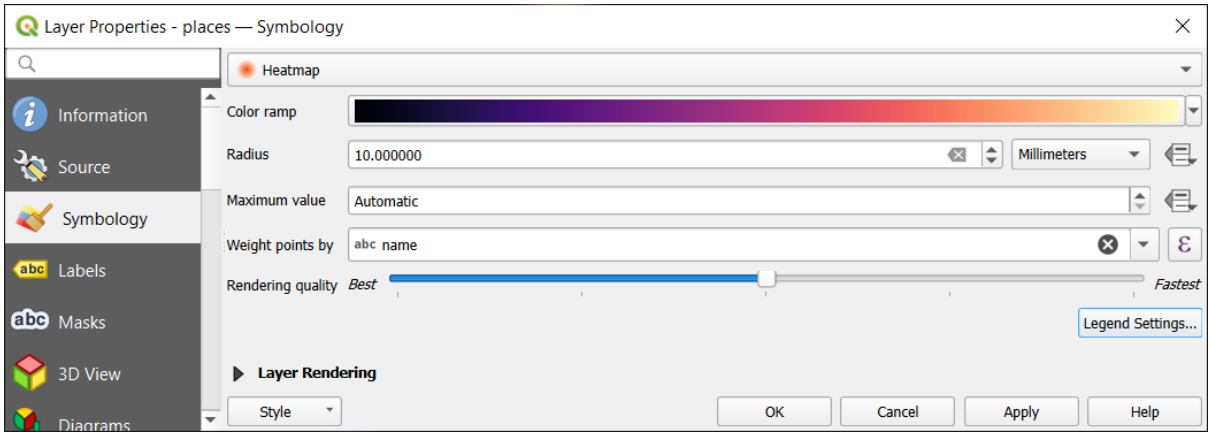
12.10 Fig. : Inverted Polygon dialog

Heatmap Renderer

With the *Heatmap* renderer you can create live dynamic heatmaps for (multi)point layers. You can specify the heatmap *Radius* in millimeters, points, pixels, map units or inches, choose and edit a *Color ramp* for the heatmap style and use a slider for selecting a trade-off between render speed and quality. You can also define a *Maximum value* limit and *Weight points by* using a field or an expression.

Use Data defined override to dynamically control *Radius* and *Maximum value* based on the attributes of your data. For example, the radius of a heatmap point could be determined by its population attribute, or the maximum value could be based on a temporal range.

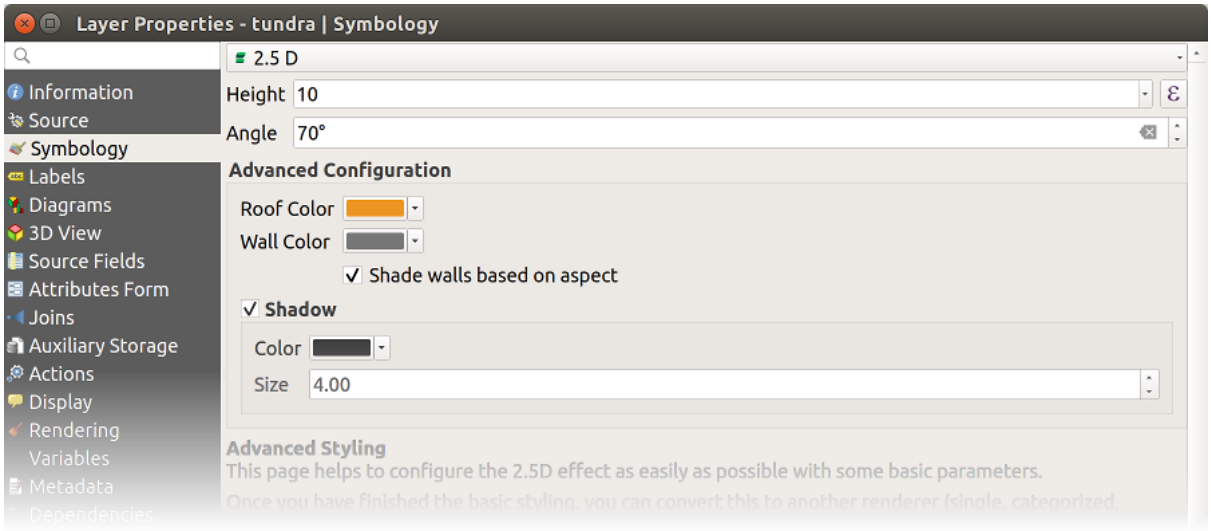
When adding or removing a feature the heatmap renderer updates the heatmap style automatically. The *Color ramp* will be shown as a legend bar and in the *Legend settings* you can set the *Labels* for the *Maximum* and *Minimum* values. You can also change the orientation and direction of the legend in the *Layout*.



12.11 Fig. : Heatmap dialog

2.5D Renderer

Using the *2.5D* renderer it's possible to create a 2.5D effect on your layer's features. You start by choosing a *Height* value (in map units). For that you can use a fixed value, one of your layer's fields, or an expression. You also need to choose an *Angle* (in degrees) to recreate the viewer position (0° means west, growing in counter clock wise). Use advanced configuration options to set the *Roof Color* and *Wall Color*. If you would like to simulate solar radiation on the features walls, make sure to check the *Shade walls based on aspect* option. You can also simulate a shadow by setting a *Color* and *Size* (in map units).



12.12 Fig. : 2.5D dialog

Patarimas

Using 2.5D effect with other renderers

Once you have finished setting the basic style on the 2.5D renderer, you can convert this to another renderer (single, categorized, graduated). The 2.5D effects will be kept and all other renderer specific options will be available for you to fine tune them (this way you can have for example categorized symbols with a nice 2.5D representation or add some extra styling to your 2.5D symbols). To make sure that the shadow and the „building“

itself do not interfere with other nearby features, you may need to enable Symbols Levels (*Advanced* ► *Symbol levels…*). The 2.5D height and angle values are saved in the layer's variables, so you can edit it afterwards in the variables tab of the layer's properties dialog.

Embedded Renderer

The *Embedded Symbols* renderer allows to display the ‚native' symbology of a provided datasource. This is mostly the case with `KML` and `TAB` datasets that have predefined symbology.

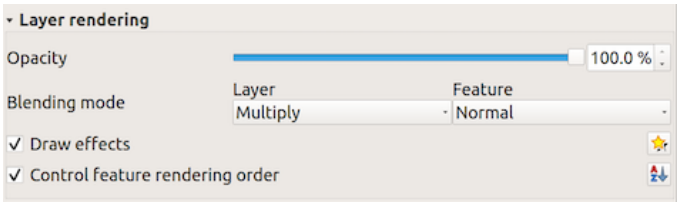
Layer rendering

From the Symbology tab, you can also set some options that invariably act on all features of the layer:

- *Opacity* : You can make the underlying layer in the map canvas visible with this tool. Use the slider to adapt the visibility of your vector layer to your needs. You can also make a precise definition of the percentage of visibility in the menu beside the slider.
- *Blending mode* at the *Layer* and *Feature* levels: You can achieve special rendering effects with these tools that you may previously only know from graphics programs. The pixels of your overlaying and underlaying layers are mixed through the settings described in *Suliejimo režimai*.
- Pritaikyti *paišymo efektus* visiems sluoksnio geoobjektams mygtuku *Paišyti efektus*.
- *Control feature rendering order* allows you, using features attributes, to define the z-order in which they shall be rendered. Activate the checkbox and click on the button beside. You then get the *Define Order* dialog in which you:

  1. Choose a field or build an expression to apply to the layer features.
  2. Set in which order the fetched features should be sorted, i.e. if you choose **Ascending** order, the features with lower value are rendered under those with higher value.
  3. Define when features returning NULL value should be rendered: **first** (bottom) or **last** (top).
  4. Repeat the above steps as many times as rules you wish to use.

  The first rule is applied to all the features in the layer, z-ordering them according to their returned value. Then, within each group of features with the same value (including those with NULL value) and thus the same z-level, the next rule is applied to sort them. And so on…



12.13 Fig. : Layer rendering options

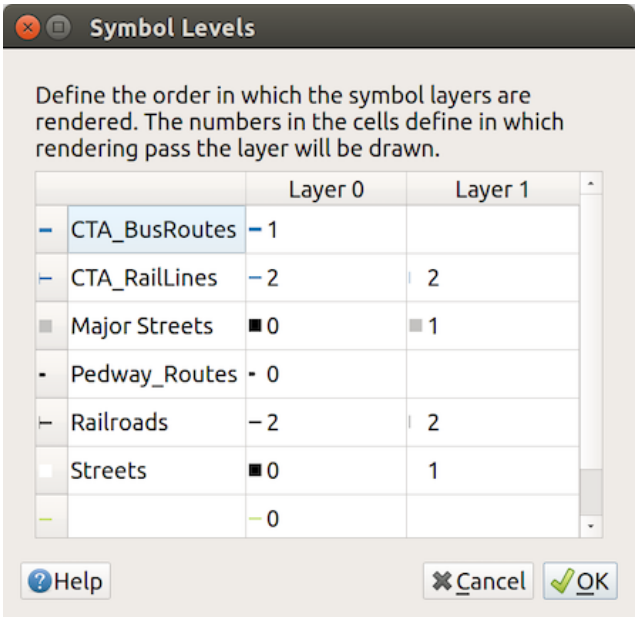
Other Settings

Symbol levels

For renderers that allow stacked symbol layers (only heatmap doesn't) there is an option to control the rendering order of each symbol's levels.

For most of the renderers, you can access the Symbols levels option by clicking the *Advanced* button below the saved symbols list and choosing *Symbol levels*. For the *Rule-based Renderer* the option is directly available through *Symbols Levels…* button, while for *Point displacement Renderer* renderer the same button is inside the *Rendering settings* dialog.

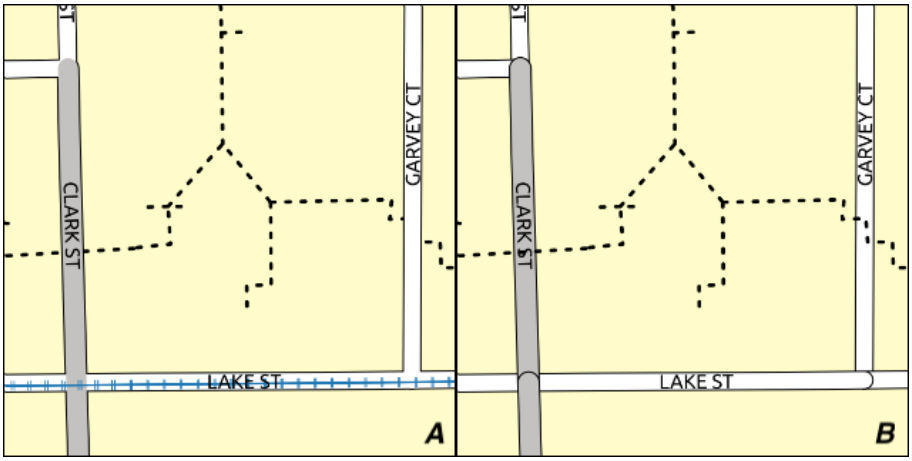
To activate symbols levels, select the *Enable symbol levels*. Each row will show up a small sample of the combined symbol, its label and the individual symbols layer divided into columns with a number next to it. The numbers represent the rendering order level in which the symbol layer will be drawn. Lower values levels are drawn first, staying at the bottom, while higher values are drawn last, on top of the others.



12.14 Fig. : Symbol levels dialog

Pastaba

If symbols levels are deactivated, the complete symbols will be drawn according to their respective features order. Overlapping symbols will simply obfuscate to other below. Besides, similar symbols won't „merge“ with each other.



12.15 Fig. : Symbol levels activated (A) and deactivated (B) difference

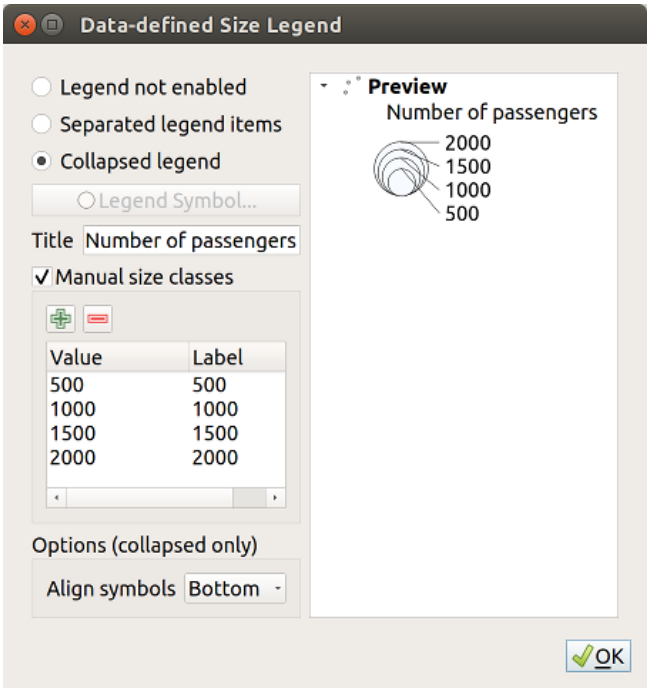
Data-defined size legend

When a layer is rendered with the *proportional symbol or the multivariate rendering* or when a *scaled size diagram* is applied to the layer, you can allow the display of the scaled symbols in both the *Layers panel* and the *print layout legend*.

To enable the *Data-defined Size Legend* dialog to render symbology, select the eponym option in the *Advanced* button below the saved symbols list. For diagrams, the option is available under the *Legend* tab. The dialog provides the following options to:

- select the type of legend: *Legend not enabled*, *Separated legend items* and *Collapsed legend*. For the latter option, you can select whether the legend items are aligned at the **Bottom** or at the **Center**;
- preview the *symbol to use* for legend representation;
- insert the title in the legend;
- resize the classes to use: by default, QGIS provides you with a legend of five classes (based on natural pretty breaks) but you can apply your own classification using the *Manual size classes* option. Use the and buttons to set your custom classes values and labels.
- For collapsed legend, it's possible to:
  - *Align symbols* in the center or the bottom
  - configure the horizontal leader *Line symbol* from the symbol to the corresponding legend text.

A preview of the legend is displayed in the right panel of the dialog and updated as you set the parameters.



12.16 Fig. : Setting size scaled legend

Pastaba

Currently, data-defined size legend for layer symbology can only be applied to point layer using single, categorized or graduated symbology.

Animation settings

To allow any symbol to become an *animated symbol*, you can utilize *Animation settings* panel. In this panel, you can enable animation for the symbol and set a specific frame rate for the symbol's redrawing.

1. Start by going to the top symbol level and select *Advanced* menu in the bottom right of the dialog
2. Find *Animation settings* option
3. Check *Is Animated* to enable animation for the symbol
4. Configure the *Frame rate*, i.e. how fast the animation would be played
5. You can now use `@symbol_frame` variable in any sub-symbol data defined property in order to animate that property.

For example, setting the symbol's rotation to data defined expression `@symbol_frame % 360` will cause the symbol to rotate over time, with rotation speed dictated by the symbol's frame rate:

Extent buffer

You may set an extent buffer for a *symbol*. This means that a buffer is applied to the current map extent so that if a feature is outside of the actual map extent but inside the buffered extent it will still be rendered. This is useful for example with symbols which use the geometry generator where you would like to still see the generated geometries even if the actual feature is outside of the map extent.

To edit the extent buffer you can utilize the *Extent buffer* panel.

1. Start by going to the top symbol level and select *Advanced* menu in the bottom right of the dialog
2. Find *Extent buffer* option
3. In the new panel you can set the buffer distance

The buffer distance units can be changed. You can also control the distance value by using the data defined override widget. For example you can change the value based on the current map scale `if(@map_scale > 50000, 5000, 0)`:

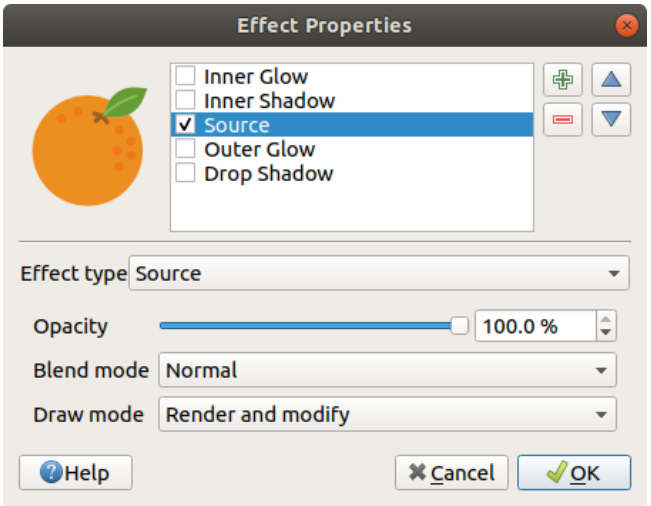
Draw effects

In order to improve layer rendering and avoid (or at least reduce) the resort to other software for final rendering of maps, QGIS provides another powerful functionality: the *Draw Effects* options, which adds paint effects for customizing the visualization of vector layers.

The option is available in the *Layer Properties* ► *Symbology* dialog, under the *Layer rendering* group (applying to the whole layer) or in *symbol layer properties* (applying to corresponding features). You can combine both usage.

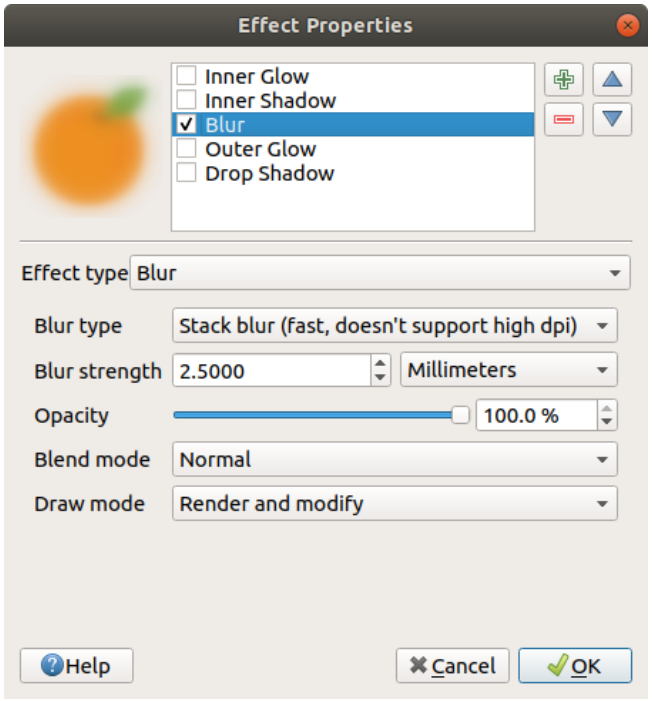
Paint effects can be activated by checking the *Draw effects* option and clicking the Customize effects button. That will open the *Effect Properties* Dialog (see Fig. 12.17). The following effect types, with custom options are available:

- **Source**: Draws the feature's original style according to the configuration of the layer's properties. The *Opacity* of its style can be adjusted as well as the *Blend mode* and *Draw mode*. These are common properties for all types of effects.



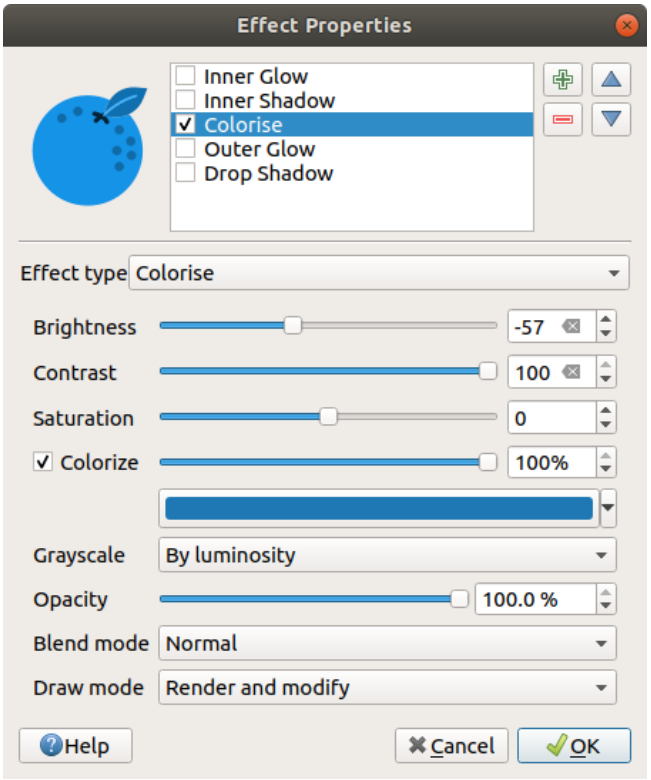
12.17 Fig. : Draw Effects: Source dialog

- **Blur**: Adds a blur effect on the vector layer. The custom options that you can change are the *Blur type* (*Stack blur (fast)* or *Gaussian blur (quality)*) and the *Blur strength*.



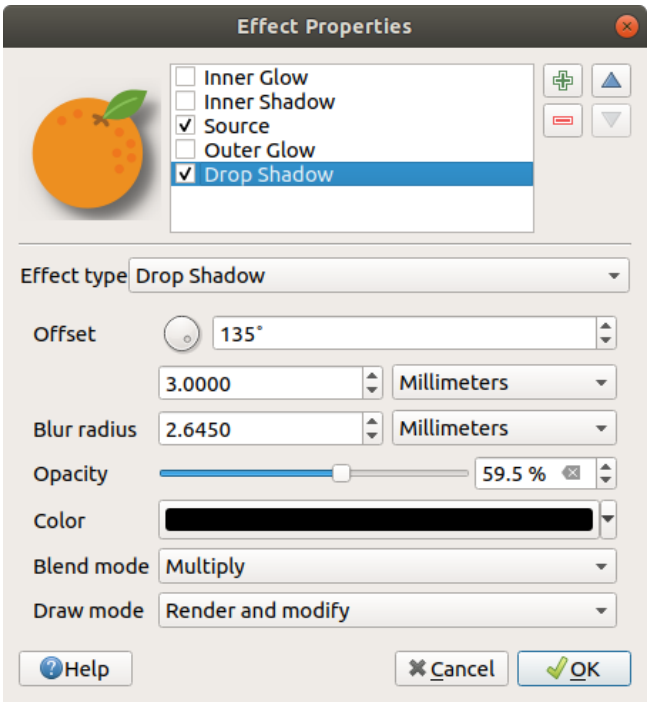
12.18 Fig. : Draw Effects: Blur dialog

- **Colorise**: This effect can be used to make a version of the style using one single hue. The base will always be a grayscale version of the symbol and you can:
  - Use the *Grayscale* to select how to create it: options are ‚By lightness', ‚By luminosity', ‚By average' and ‚Off'.
  - If *Colorise* is selected, it will be possible to mix another color and choose how strong it should be.
  - Control the *Brightness*, *Contrast* and *Saturation* levels of the resulting symbol.



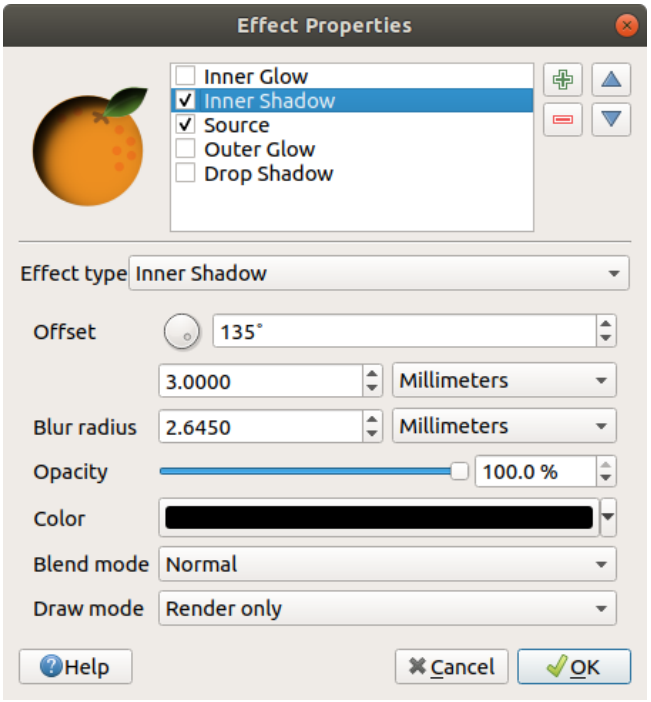
12.19 Fig. : Draw Effects: Colorize dialog

- **Drop Shadow**: Using this effect adds a shadow on the feature, which looks like adding an extra dimension. This effect can be customized by changing the *Offset* angle and distance, determining where the shadow shifts towards to and the proximity to the source object. *Drop Shadow* also has the option to change the *Blur radius* and the *Color* of the effect.



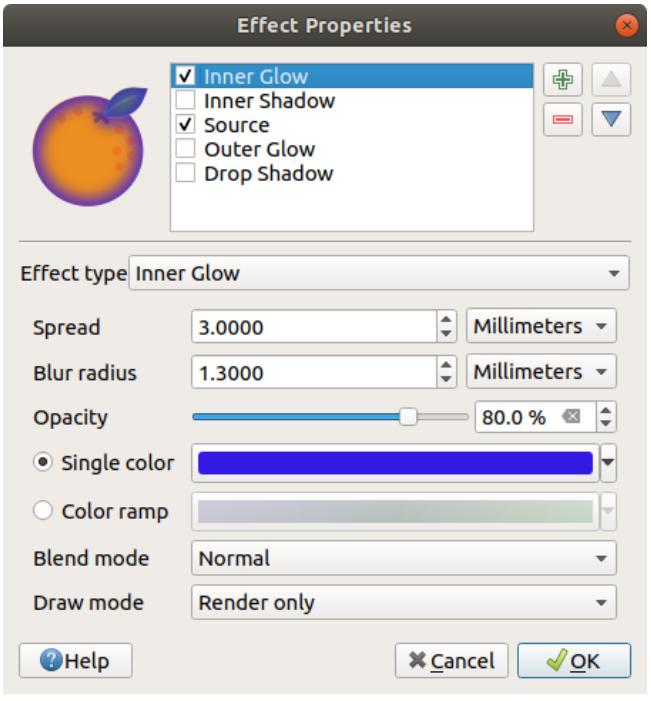
12.20 Fig. : Draw Effects: Drop Shadow dialog

- **Inner Shadow**: This effect is similar to the *Drop Shadow* effect, but it adds the shadow effect on the inside of the edges of the feature. The available options for customization are the same as the *Drop Shadow* effect.



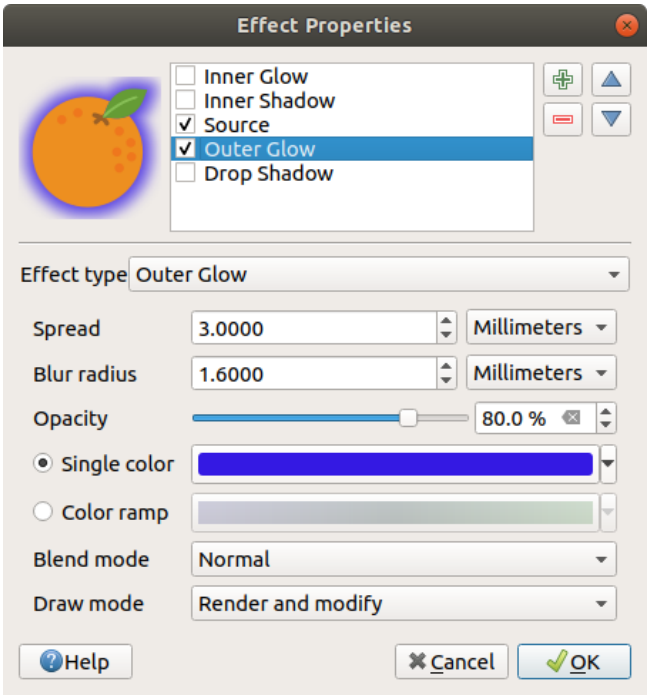
12.21 Fig. : Draw Effects: Inner Shadow dialog

- **Inner Glow**: Adds a glow effect inside the feature. This effect can be customized by adjusting the *Spread* (width) of the glow, or the *Blur radius*. The latter specifies the proximity from the edge of the feature where you want any blurring to happen. Additionally, there are options to customize the color of the glow using a *Single color* or a *Color ramp*.



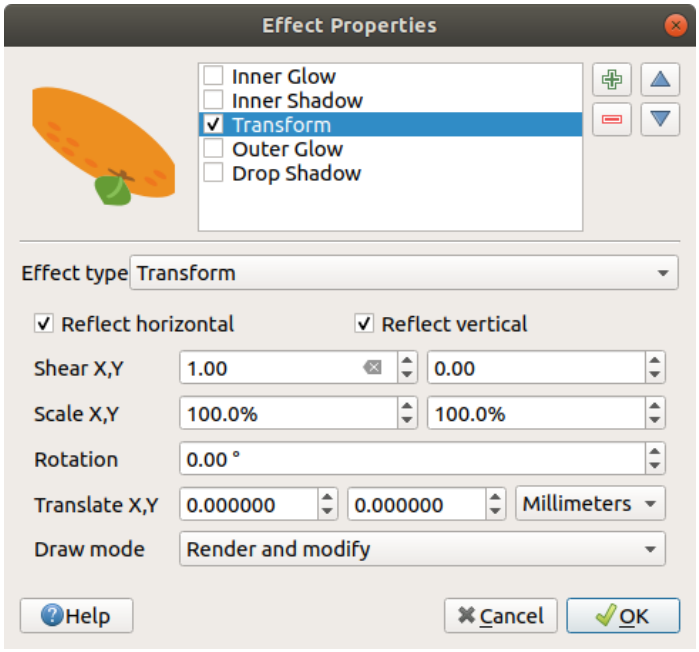
12.22 Fig. : Draw Effects: Inner Glow dialog

- **Outer Glow**: This effect is similar to the *Inner Glow* effect, but it adds the glow effect on the outside of the edges of the feature. The available options for customization are the same as the *Inner Glow* effect.



12.23 Fig. : Draw Effects: Outer Glow dialog

- **Transform**: Adds the possibility of transforming the shape of the symbol. The first options available for customization are the *Reflect horizontal* and *Reflect vertical*, which actually create a reflection on the horizontal and/or vertical axes. The other options are:
  - *Shear X,Y*: Slants the feature along the X and/or Y axis.
  - *Scale X,Y*: Enlarges or minimizes the feature along the X and/or Y axis by the given percentage.
  - *Rotation*: Turns the feature around its center point.
  - and *Translate X,Y* changes the position of the item based on a distance given on the X and/or Y axis.



12.24 Fig. : Draw Effects: Transform dialog

One or more effect types can be used at the same time. You (de)activate an effect using its checkbox in the effects list. You can change the selected effect type by using the *Effect type* option. You can reorder the effects using Move up and Move down buttons, and also add/remove effects using the Add new effect and Remove effect buttons.

There are some common options available for all draw effect types. *Opacity* and *Blend mode* options work similar to the ones described in *Layer rendering* and can be used in all draw effects except for the transform one.

There is also a *Draw mode* option available for every effect, and you can choose whether to render and/or modify the symbol, following some rules:

- Effects render from top to bottom.
- *Render only* mode means that the effect will be visible.
- *Modifier only* mode means that the effect will not be visible but the changes that it applies will be passed to the next effect (the one immediately below).
- The *Render and Modify* mode will make the effect visible and pass any changes to the next effect. If the effect is at the top of the effects list or if the immediately above effect is not in modify mode, then it will use the original source symbol from the layers properties (similar to source).

12.1.4 Labels Properties

The *Labels* properties provides you with all the needed and appropriate capabilities to configure smart labeling on vector layers. This dialog can also be accessed from the *Layer Styling* panel, or using the Layer Labeling Options button of the **Labels toolbar**.

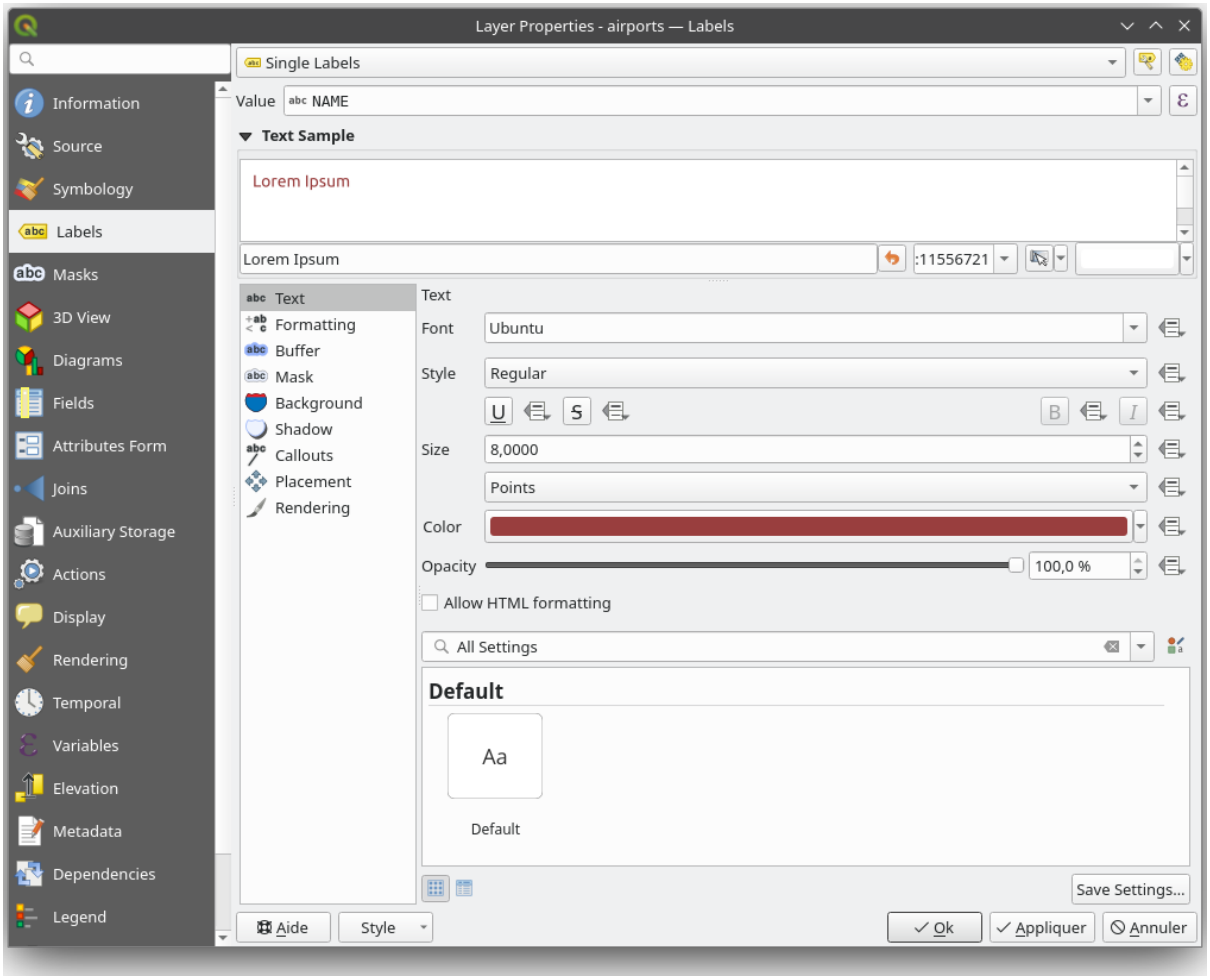
At the top of the dialog, you have:

- a combobox for selecting the appropriate labeling method for the active layer
- the Configure project labeling rules button: helps you control interactions between labels and features across the layers in the project. More details at *Configuring project labeling rules*.
- the Automated placement settings (applies to all layers) button: configure general properties on label placement and conflicts resolution. More details at *Setting the automated placement engine*.

The first step is to choose the labeling method from the drop-down list. Available methods are:

- *No labels*: the default value, showing no labels from the layer
- *Single labels*: Show labels on the map using a single attribute or an expression
- *Rule-based labeling*
- and *Blocking*: allows to set a layer as just an obstacle for other layer's labels without rendering any labels of its own.

The next steps assume you select the *Single labels* option, opening the following dialog.



12.25 Fig. : Layer labeling settings - Single labels

At the top of the dialog, a *Value* drop-down list is enabled. You can select an attribute column to use for labeling. By default, the *display field* is used. Click ε if you want to define labels based on expressions - See *Define labels based on expressions*.

Pastaba

Labels with their formatting can be displayed as entries in the legends, if enabled in the *Legend* tab.

Below are displayed options to customize the labels, under various tabs:

- abc *Text*

- *Formatting*
- *Buffer*
- *Mask*
- *Background*
- *Shadow*
- *Callouts*
- *Placement*
- *Rendering*

Description of how to set each property is exposed at *Užrašo nustatymas*.

Configuring project labeling rules

Pressing the Configure project labeling rules button next to the labeling method drop-down selector, you can create rules that controls how labels from a layer can interact with labels or features from another layer.

1. Press the Add rule button and in the drop-down menu, select one of the rule types:
   - *Prevent labels overlapping features*: prevents labels being placed overlapping features from a different layer.
   - *Pull labels towards features*: prevents labels being placed too far from features from a different layer. The maximum distance can be set in the unit of your choice.
   - *Push labels away from features*: prevents labels being placed too close to features from a different layer. The minimum distance can be set in the unit of your choice, as well as the *rule's priority* (The highest-priority rules are more important to respect in the event of a label placement conflict).
   - *Push labels away from other labels*: prevents labels being placed too close to labels from a different layer.

   Dėmesio

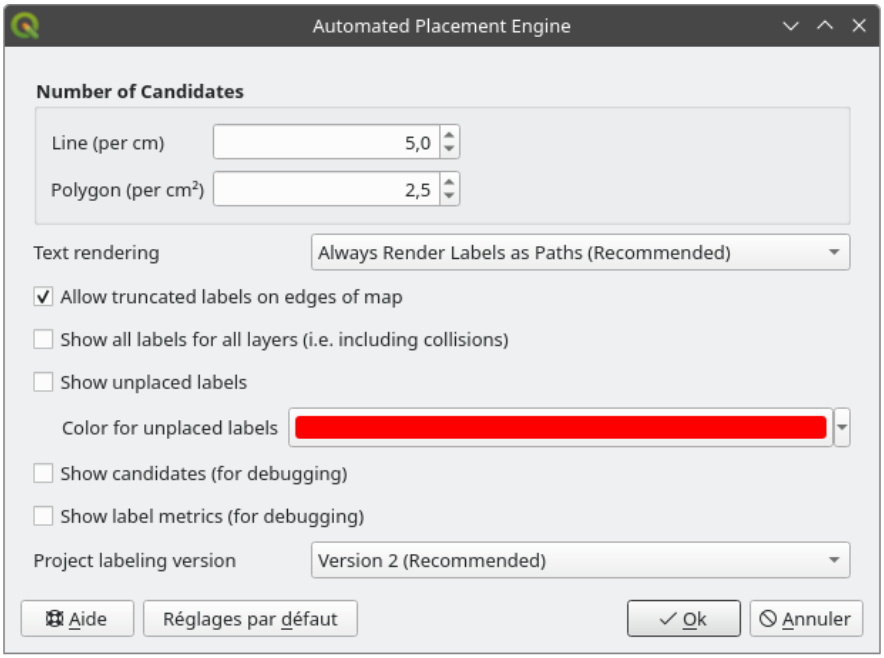
   The last three options are only available on QGIS installed with GEOS >= 3.10 (see *Help* ► *About* menu).

2. Fill the properties at your will; you can provide a more meaningful name to the rule.
3. Press *OK*.
4. Add as many rules as necessary.
5. If necessary, press Edit rule to modify the selected rule or Remove rule to delete it from the project.

The set rules are available from any layer *Labels* properties tab, pressing the Configure project labeling rules button. You can temporary enable or disable any of them, using the checkbox next to the name. Hover over a rule to preview its details.

Setting the automated placement engine

You can use the automated placement settings to configure a project-level automated behavior of the labels. In the top right corner of the *Labels* tab, click the Automated placement settings (applies to all layers) button, opening a dialog with the following options:



12.26 Fig. : The labels automated placement engine

- *Number of candidates*: calculates and assigns to line and polygon features the number of possible labels placement based on their size. The longer or wider a feature is, the more candidates it has, and its labels can be better placed with less risk of collision.
- *Text rendering*: sets the default value for label rendering widgets when *exporting a map canvas* or *a layout* to PDF or SVG. If *Always render labels as text* is selected then labels can be edited in external applications (e.g. Inkscape) as normal text. BUT the side effect is that the rendering quality is decreased, and there are issues with rendering when certain text settings like buffers are in place. That's why *Always render labels as paths (recommended)* which exports labels as outlines but guarantees complete compatibility with the full range of formatting options available, is recommended. With *Prefer rendering labels as text* option, labels are rendered as text objects, unless doing so results in rendering artifacts or poor quality rendering (depending on text format settings).

> **Pastaba**
>
> When rendering labels as text to a vector based device (e.g. PDF or SVG), care must be taken to ensure that the required fonts are available to users when opening the created files, or default fallback fonts will be used to display the output instead. (Although PDF exports MAY automatically embed some fonts when possible, depending on the user's platform).

- *Allow truncated labels on edges of map*: controls whether labels which fall partially outside of the map extent should be rendered. If checked, these labels will be shown (when there's no way to place them fully within the visible area). If unchecked then partially visible labels will be skipped. Note that this setting has no effects on labels' display in the *layout map item*.
- *Show all labels for all layers (i.e. including colliding objects)*. Note that this option can be also set per layer (see *Braižymo kortelė*)

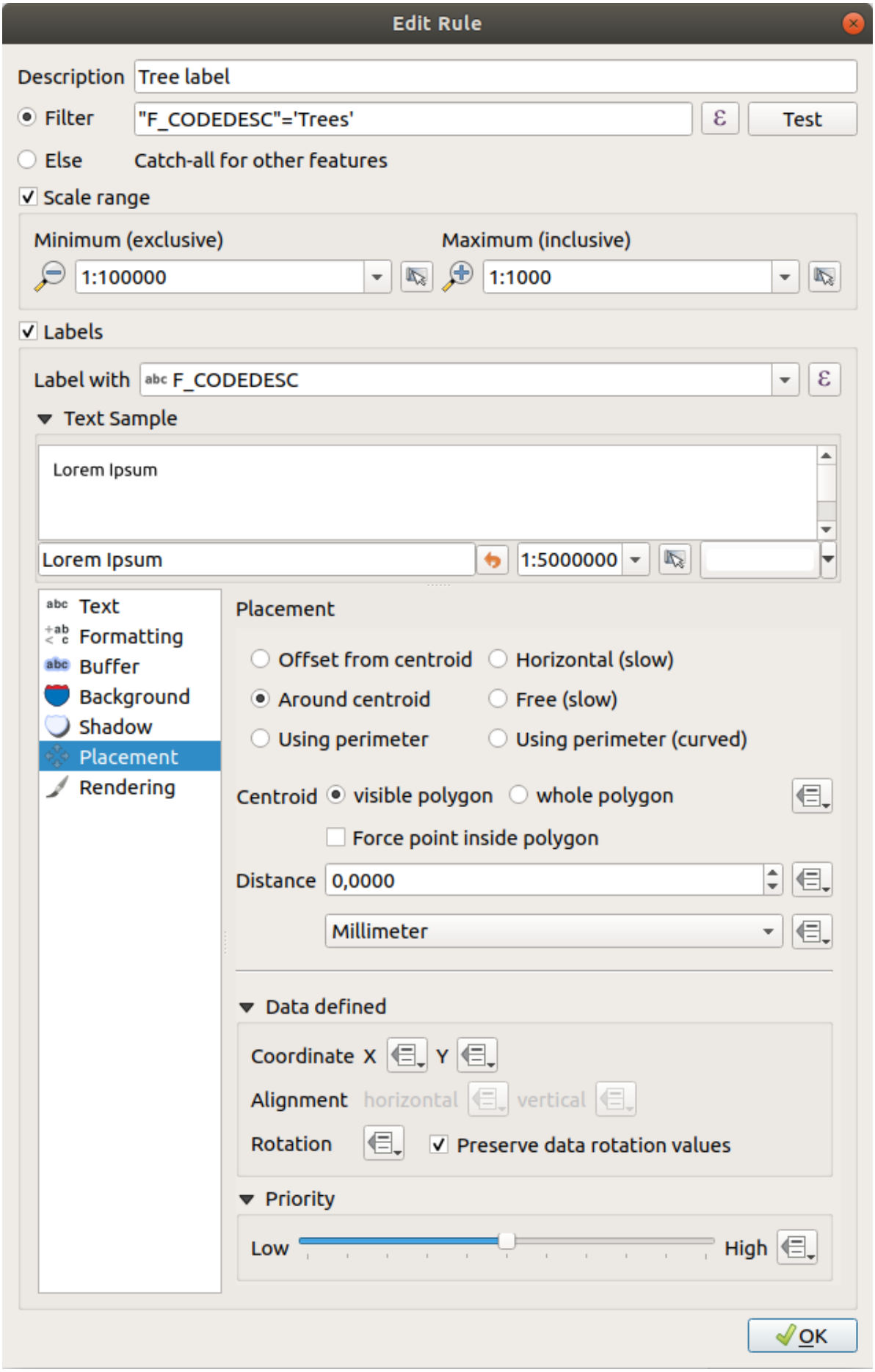
- *Show unplaced labels*: allows to determine whether any important labels are missing from the maps (e.g. due to overlaps or other constraints). They are displayed using a customizable color.
- *Show candidates (for debugging)*: controls whether boxes should be drawn on the map showing all the candidates generated for label placement. Like the label says, it's useful only for debugging and testing the effect different labeling settings have. This could be handy for a better manual placement with tools from the *label toolbar*.
- *Show label metrics (for debugging)*: displays the text bounds of the label in red and baselines in blue
- *Project labeling version*: QGIS supports two different versions of label automatic placement:
  - *Version 1*: the old system (used by QGIS versions 3.10 and earlier, and when opening projects created in these versions in QGIS 3.12 or later). Version 1 treats label and obstacle priorities as „rough guides" only, and it's possible that a low-priority label will be placed over a high-priority obstacle in this version. Accordingly, it can be difficult to obtain the desired labeling results when using this version and it is thus recommended only for compatibility with older projects.
  - *Version 2 (recommended)*: this is the default system in new projects created in QGIS 3.12 or later. In version 2, the logic dictating when labels are allowed to overlap *obstacles* has been reworked. The newer logic forbids any labels from overlapping any obstacles with a greater obstacle weight compared to the label's priority. As a result, this version results in much more predictable and easier to understand labeling results.

Rule-based labeling

With rule-based labeling multiple label configurations can be defined and applied selectively on the base of expression filters and scale range, as in *Rule-based rendering*.

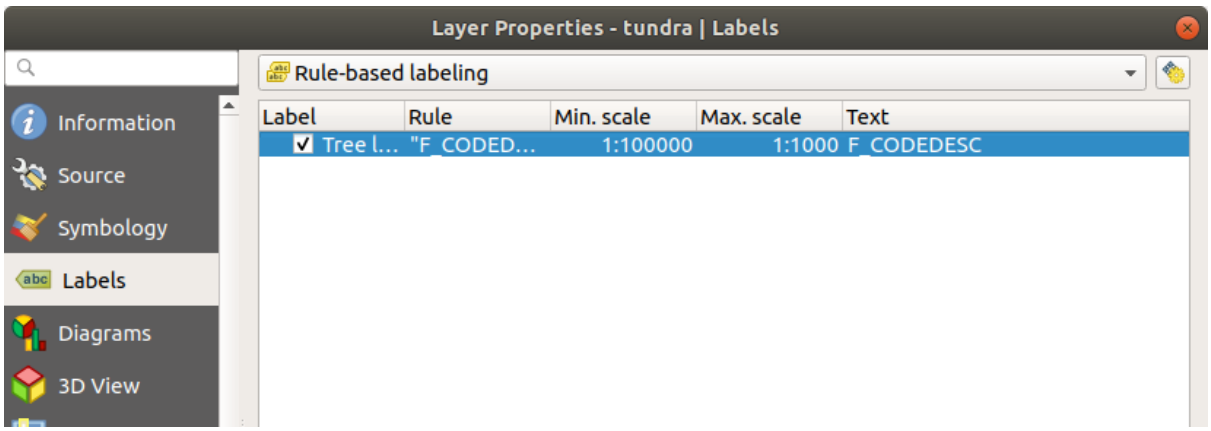
To create a rule:

1. Select the **Rule-based labeling** option in the main drop-down list from the *Labels* tab
2. Click the Add rule button at the bottom of the dialog.
3. Fill the new dialog with:
   - *Description*: a text used to identify the rule in the *Labels* tab and as a *label legend entry* in the print layout legend
   - *Filter*: an expression to select the features to apply the label settings to
   - If there are rules already set, the *Else* option can be used to select all the features not matching any filter of the rules in the same group.
4. You can set a *scale range* in which the label rule should be applied.
5. The options available under the *Labels* group box are the usual *label settings*. Configure them and press *OK*.



12.27 Fig. : Rule settings

A summary of existing rules is shown in the main dialog (see Fig. 12.28). You can add multiple rules, reorder or imbricate them with a drag-and-drop. You can as well remove them with the button or edit them with button or a double-click.



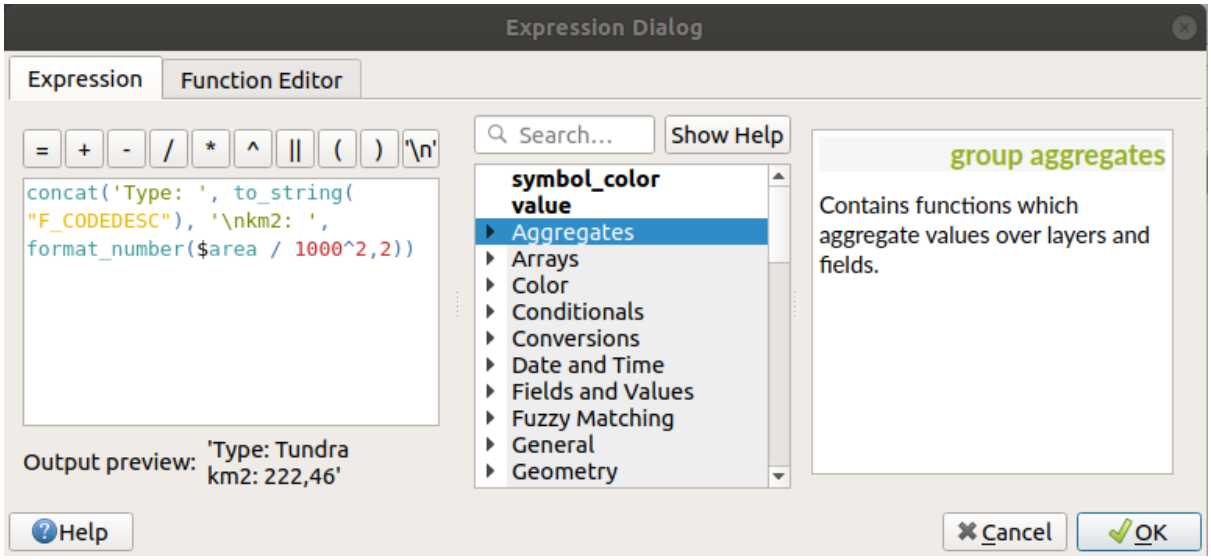
12.28 Fig. : Rule based labeling panel

Define labels based on expressions

Whether you choose single or rule-based labeling type, QGIS allows using expressions to label features.

Assuming you are using the *Single labels* method, click the button near the *Value* drop-down list in the *Labels* tab of the properties dialog.

In Fig. 12.29, you see a sample expression to label the alaska trees layer with tree type and area, based on the field ‚VEGDESC', some descriptive text, and the function `$area` in combination with `format_number()` to make it look nicer.



12.29 Fig. : Using expressions for labeling

Expression based labeling is easy to work with. All you have to take care of is that:

- You may need to combine all elements (strings, fields, and functions) with a string concatenation function such as `concat`, `+` or `||`. Be aware that in some situations (when null or numeric value are involved) not all of these tools will fit your need.

- Strings are written in ‚single quotes'.
- Fields are written in „double quotes" or without any quote.

Let's have a look at some examples:

1. Label based on two fields ‚name' and ‚place' with a comma as separator:

```
"name" || ', ' || "place"
```

   Returns:

```
John Smith, Paris
```

2. Label based on two fields ‚name' and ‚place' with other texts:

```
'My name is ' + "name" + 'and I live in ' + "place"
'My name is ' || "name" || 'and I live in ' || "place"
concat('My name is ', name, ' and I live in ', "place")
```

   Returns:

```
My name is John Smith and I live in Paris
```

3. Label based on two fields ‚name' and ‚place' with other texts combining different concatenation functions:

```
concat('My name is ', name, ' and I live in ' || place)
```

   Returns:

```
My name is John Smith and I live in Paris
```

   Or, if the field ‚place' is NULL, returns:

```
My name is John Smith
```

4. Multi-line label based on two fields ‚name' and ‚place' with a descriptive text:

```
concat('My name is ', "name", '\n' , 'I live in ' , "place")
```

   Returns:

```
My name is John Smith
I live in Paris
```

5. Label based on a field and the $area function to show the place's name and its rounded area size in a converted unit:

```
'The area of ' || "place" || ' has a size of '
|| round($area/10000) || ' ha'
```

   Returns:

```
The area of Paris has a size of 10500 ha
```

6. Create a CASE ELSE condition. If the population value in field *population* is <= 50000 it is a town, otherwise it is a city:

```
concat('This place is a ',
CASE WHEN "population" <= 50000 THEN 'town' ELSE 'city' END)
```

   Returns:

```
This place is a town
```

7. Display name for the cities and no label for the other features (for the „city“ context, see example above):

```
CASE WHEN "population" > 50000 THEN "NAME" END
```

Returns:

```
Paris
```

As you can see in the expression builder, you have hundreds of functions available to create simple and very complex expressions to label your data in QGIS. See *Išraiškos* chapter for more information and examples on expressions.

Using data-defined override for labeling

With the Data defined override function, the settings for the labeling are overridden by entries in the attribute table or expressions based on them. This feature can be used to set values for most of the labeling options described above.

For example, using the Alaska QGIS sample dataset, let's label the `airports` layer with their name, based on their militarian `USE`, i.e. whether the airport is accessible to :

- military people, then display it in gray color, size 8;
- others, then show in blue color, size 10.

To do this, after you enabled the labeling on the `NAME` field of the layer (see *Užrašo nustatymas*):

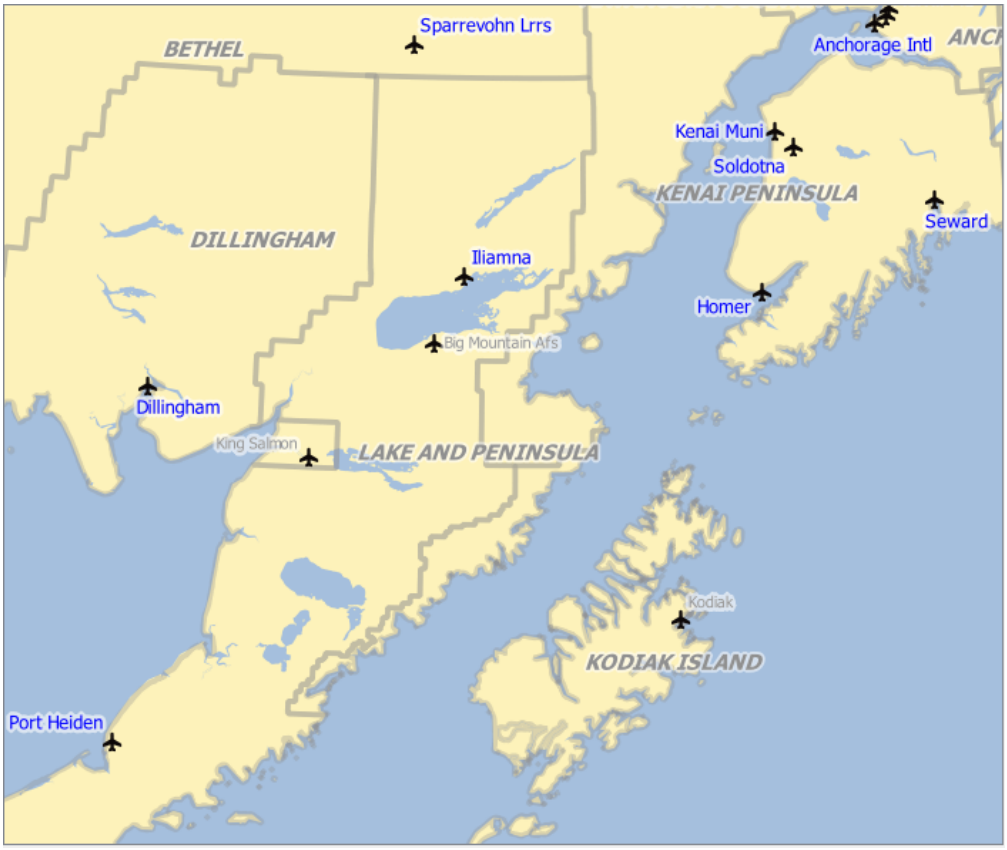
1. Activate the *Text* tab.
2. Click on the icon next to the *Size* property.
3. Select *Edit...* and type:

```
CASE
  WHEN "USE" like '%Military%' THEN 8 -- because compatible values are
↪'Military'
                                      -- and 'Joint Military/Civilian'
  ELSE 10
END
```

4. Press *OK* to validate. The dialog closes and the button becomes meaning that an rule is being run.
5. Then click the button next to the color property, type the expression below and validate:

```
CASE
  WHEN "USE" like '%Military%' THEN '150, 150, 150'
  ELSE '0, 0, 255'
END
```

Likewise, you can customize any other property of the label, the way you want. See more details on the Data-define override widget's description and manipulation in *Nuo duomenų priklausančių permušimų nustatymai* section.



12.30 Fig. : Airports labels are formatted based on their attributes

Patarimas

Use the data-defined override to label every part of multi-part features

There is an option to set the labeling for multi-part features independently from your label properties. Choose the *Rendering*, `Feature options`, go to the Data-define override button next to the checkbox *Label every part of multipart-features* and define the labels as described in *Nuo duomenų priklausančių permušimų nustatymai*.

The Label Toolbar

The *Label Toolbar* provides some tools to manipulate *label* (including their *callout*) or *diagram* properties:

12.31 Fig. : The Label toolbar

- Highlight Pinned Labels, Diagrams and Callouts. If the vector layer of the item is editable, then the highlighting is green, otherwise it's blue.
- Toggle Display of Unplaced Labels: Allows to determine whether any important labels are missing from the maps (e.g. due to overlaps or other constraints). They are displayed with a customizable color (see *Setting the automated placement engine*).

- Pin/Unpin Labels and Diagrams. By clicking or draging an area, you pin overlaid items. If you click or drag an area holding `Shift`, the items are unpinned. Finally, you can also click or drag an area holding `Ctrl` to toggle their pin status.
- Show/Hide Labels and Diagrams. If you click on the items, or click and drag an area holding `Shift`, they are hidden. When an item is hidden, you just have to click on the feature to restore its visibility. If you drag an area, all the items in the area will be restored.
- Move a Label, Diagram or Callout: click to select the item and click to move it to the desired place. The new coordinates are stored in *auxiliary fields*. Selecting the item with this tool and hitting the `Delete` key will delete the stored position value.
- Rotate a Label. Click to select the label and click again to apply the desired rotation. Likewise, the new angle is stored in an auxiliary field. Selecting a label with this tool and hitting the `Delete` key will delete the rotation value of this label.
- Change Label Properties. It opens a dialog to change the clicked label properties; it can be the label itself, its coordinates, angle, font, size, multiline alignment … as long as this property has been mapped to a field. Here you can set the option to *Label every part of a feature*.

Įspėjimas

Label tools overwrite current field values

Using the *Label toolbar* to customize the labeling actually writes the new value of the property in the mapped field. Hence, be careful to not inadvertently replace data you may need later!

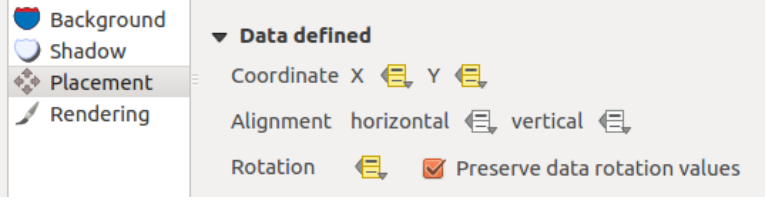
Pastaba

The *Auxiliary Storage Properties* mechanism may be used to customize labeling (position, and so on) without modifying the underlying data source.

Customize the labels from the map canvas

Combined with the *Label Toolbar*, the data defined override setting helps you manipulate labels in the map canvas (move, edit, rotate). We now describe an example using the data-defined override function for the Move Label, Diagram or Callout function (see Fig. 12.32).

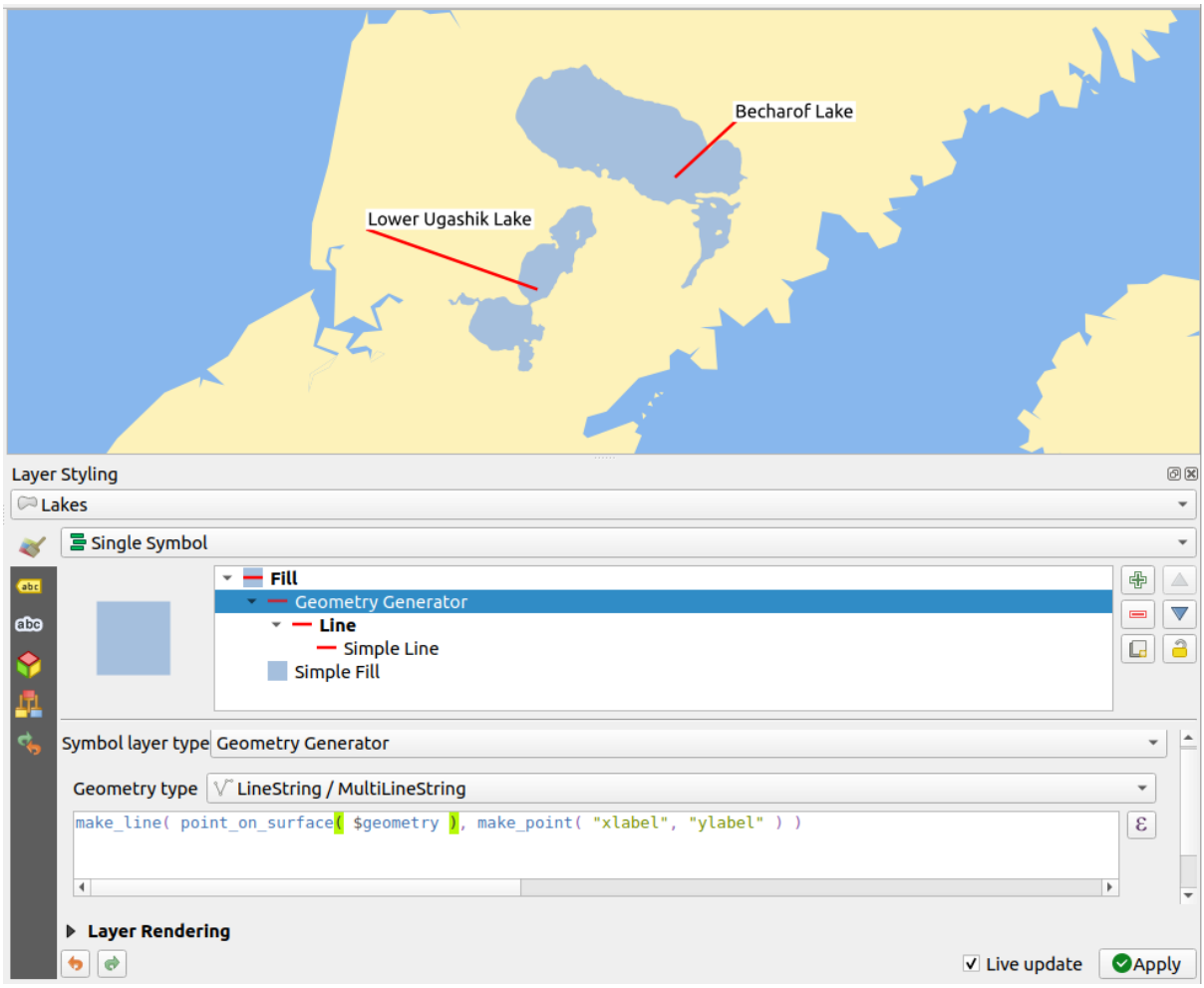
1. Import `lakes.shp` from the QGIS sample dataset.
2. Double-click the layer to open the Layer Properties. Click on *Labels* and *Placement*. Select *Offset from centroid*.
3. Look for the *Data defined* entries. Click the icon to define the field type for the *Coordinate*. Choose `xlabel` for X and `ylabel` for Y. The icons are now highlighted in yellow.



12.32 Fig. : Labeling of vector polygon layers with data-defined override

4. Zoom into a lake.
5. Set editable the layer using the Toggle Editing button.
6. Go to the Label toolbar and click the icon. Now you can shift the label manually to another position (see Fig. 12.33). The new position of the label is saved in the `xlabel` and `ylabel` columns of the attribute table.
7. It's also possible to add a line connecting each lake to its moved label using:
   - the label's *callout property*
   - or the *geometry generator symbol layer* with the expression below:

```
make_line( centroid( $geometry ), make_point( "xlabel", "ylabel" ) )
```



12.33 Fig. : Moved labels

Pastaba

The *Auxiliary Storage Properties* mechanism may be used with data-defined properties without having an editable data source.

12.1.5 Diagrams Properties

The *Diagrams* tab allows you to add a graphic overlay to a vector layer (see Fig. 12.34). This dialog can also be accessed from the *Layer Styling* panel, or using the Layer Diagram Options button of the **Labels toolbar**.

The current core implementation of diagrams provides support for:

- *No diagrams*: the default value with no diagram displayed over the features;
- *Pie chart*, a circular statistical graphic divided into slices to illustrate numerical proportion. The arc length of each slice is proportional to the quantity it represents;
- *Text diagram*, a horizontally divided circle showing statistic values inside;
- *Histogram*, bars of varying colors for each attribute aligned next to each other;
- *Stacked bars*, stacks bars of varying colors for each attribute on top of each other vertically or horizontally;
- *Stacked diagram*, stacks diagrams of equal or varying types, next to each other, vertically or horizontally. More details at *Stacked Diagrams*.

In the top right corner of the *Diagrams* tab, the Automated placement settings (applies to all layers) button provides means to control diagram *labels placement* on the map canvas.

> **Patarimas**
>
> **Switch quickly between types of diagrams**
>
> Given that the settings are almost common to the different types of diagram, when designing your diagram, you can easily change the diagram type and check which one is more appropriate to your data without any loss.

For each type of diagram, the properties are divided into several tabs:

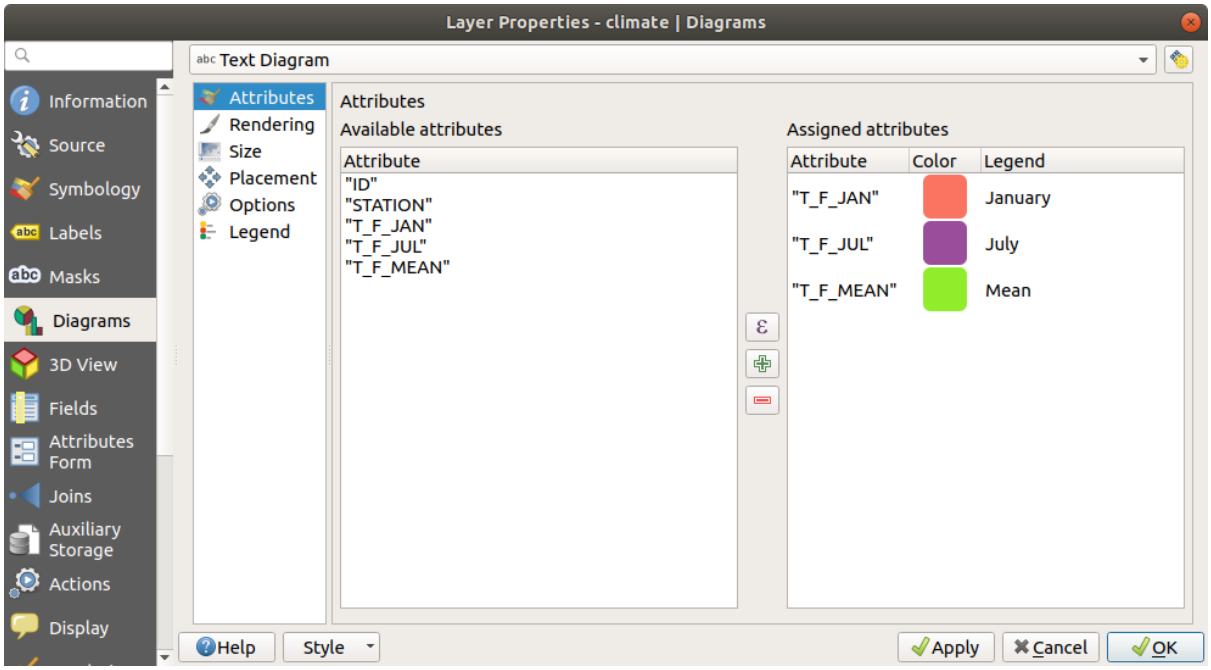
- *Attributes*
- *Rendering*
- *Size*
- *Placement*
- *Options*
- *Legend*

Attributes

Attributes defines which variables to display in the diagram. Use add item button to select the desired fields into the ‚Assigned Attributes' panel. Generated attributes with *Išraiškos* can also be used.

You can move up and down any row with click and drag, sorting how attributes are displayed. You can also change the label in the ‚Legend' column or the attribute color by double-clicking the item.

This label is the default text displayed in the legend of the print layout or of the layer tree.



12.34 Fig. : Diagram properties - Attributes tab

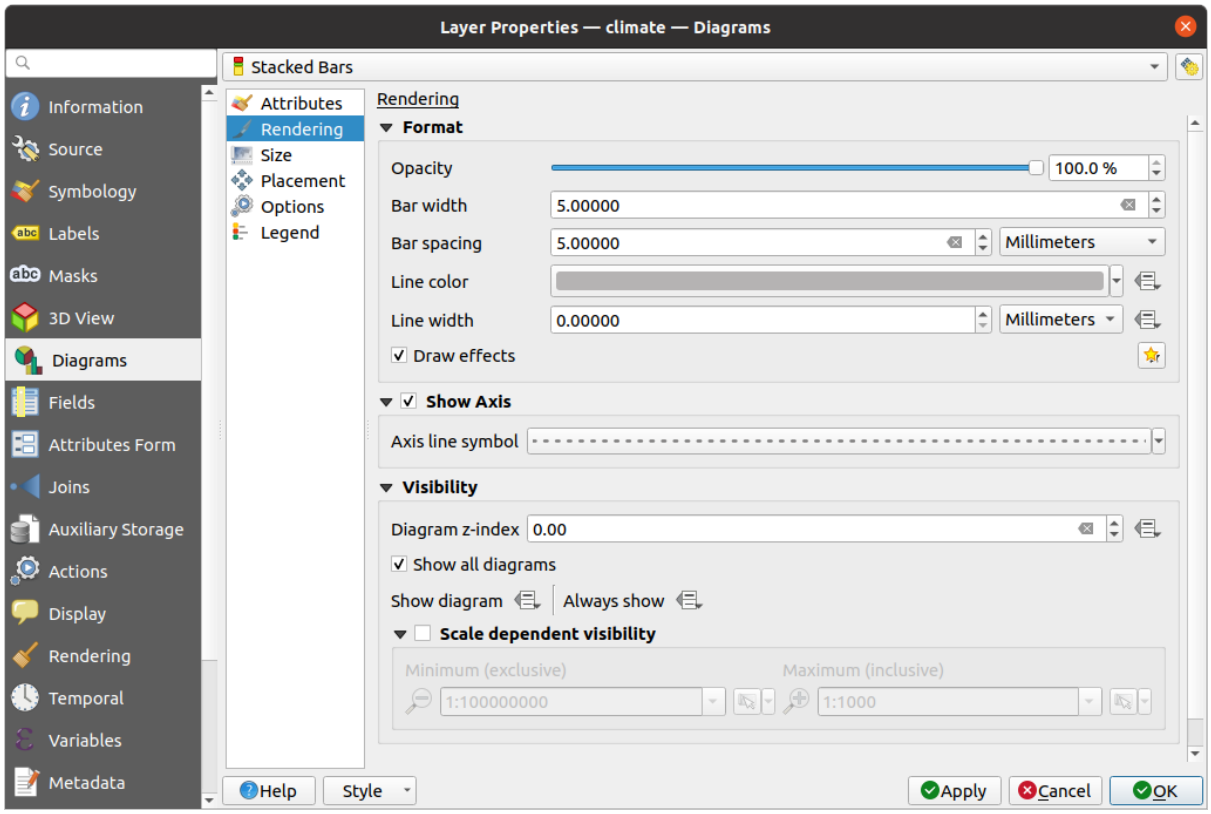
Rendering

Rendering defines how the diagram looks like. It provides general settings that do not interfere with the statistic values such as:

- the graphic's opacity, its outline width and color;
- depending on the type of diagram:
  - for histogram and stacked bars, the width of the bar and the spacing between the bars. You may want to set the spacing to `0` for stacked bars. Moreover, the *Axis line symbol* can be made visible on the map canvas and customized using *line symbol properties*.
  - for text diagram, the circle background color and the *font* used for texts;
  - for pie charts, the *Start angle* of the first slice and their *Direction* (clockwise or not).
- the use of *paint effects* on the graphics.

In this tab, you can also manage and fine tune the diagram visibility with different options:

- *Diagram z-index*: controls how diagrams are drawn on top of each other and on top of labels. A diagram with a high index is drawn over other diagrams and labels;
- *Show all diagrams*: shows all the diagrams even if they overlap each other;
- *Show diagram*: allows only specific diagrams to be rendered;
- *Always Show*: selects specific diagrams to always render, even when they overlap other diagrams or map labels;
- setting the *Scale dependent visibility*;

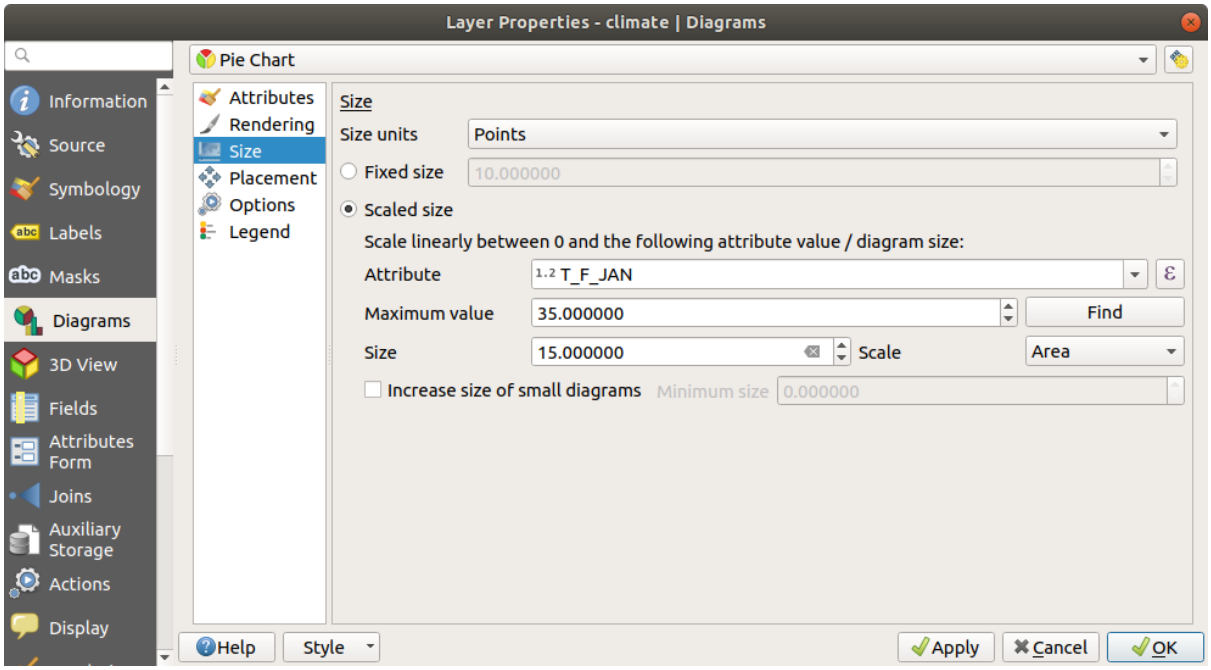


12.35 Fig. : Diagram properties - Rendering tab

Size

Size is the main tab to set how the selected statistics are represented. The diagram size *units* can be ‚Millimeters', ‚Points', ‚Pixels', ‚Map Units' or ‚Inches'. You can use:

- *Fixed size*, a unique size to represent the graphic of all the features (not available for histograms)
- or *Scaled size*, based on an expression using layer attributes:
  1. In *Attribute*, select a field or build an expression
  2. Press *Find* to return the *Maximum value* of the attribute or enter a custom value in the widget.
  3. For histogram and stacked bars, enter a *Bar length* value, used to represent the *Maximum value* of the attributes. For each feature, the bar length will then be scaled linearly to keep this matching.
  4. For pie chart and text diagram, enter a *Size* value, used to represent the *Maximum value* of the attributes. For each feature, the circle area or diameter will then be scaled linearly to keep this matching (from `0`). A *Minimum size* can however be set for small diagrams.



12.36 Fig. : Diagram properties - Size tab

Placement

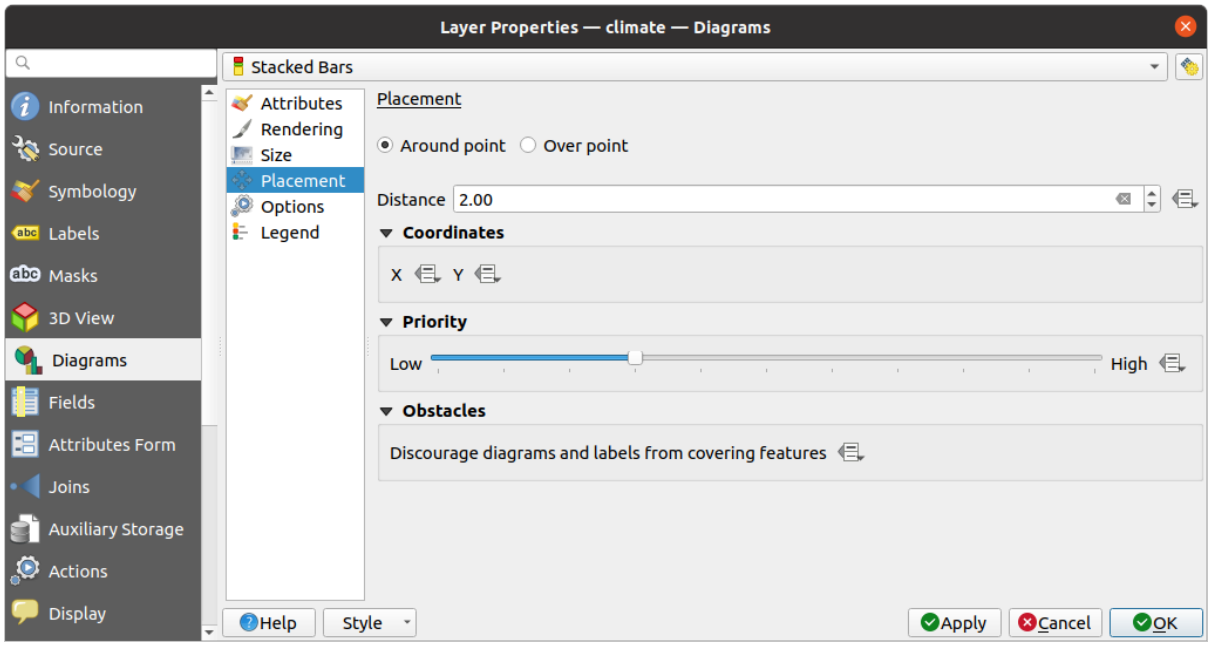
Placement defines the diagram position. Depending on the layer geometry type, it offers different options for the placement (more details at *Placement*):

- *Around point* or *Over point* for point geometry. The former variable requires a radius to follow.
- *Around line* or *Over line* for line geometry. Like point feature, the first variable requires a distance to respect and you can specify the diagram placement relative to the feature (‚above', ‚on' and/or ‚below' the line) It's possible to select several options at once. In that case, QGIS will look for the optimal position of the diagram. Remember that you can also use the line orientation for the position of the diagram.
- *Around centroid* (at a set *Distance*), *Over centroid*, *Using perimeter* and *Inside polygon* are the options for polygon features.

The *Coordinate* group provides direct control on diagram placement, on a feature-by-feature basis, using their attributes or an expression to set the *X* and *Y* coordinate. The information can also be filled using the *Move labels and diagrams* tool.

In the *Priority* section, you can define the placement priority rank of each diagram, i.e. if there are different diagrams or labels candidates for the same location, the item with the higher priority will be displayed and the others could be left out.

Discourage diagrams and labels from covering features defines features to use as *obstacles*, i.e. QGIS will try to not place diagrams nor labels over these features. The priority rank is then used to evaluate whether a diagram could be omitted due to a greater weighted obstacle feature.



12.37 Fig. : Vector properties dialog with diagram properties, Placement tab

Parinktys

The *Options* tab has settings for histograms and stacked bars. You can choose whether the *Bar orientation* should be *Up*, *Down*, *Right* or *Left*, for horizontal and vertical diagrams.

Legend

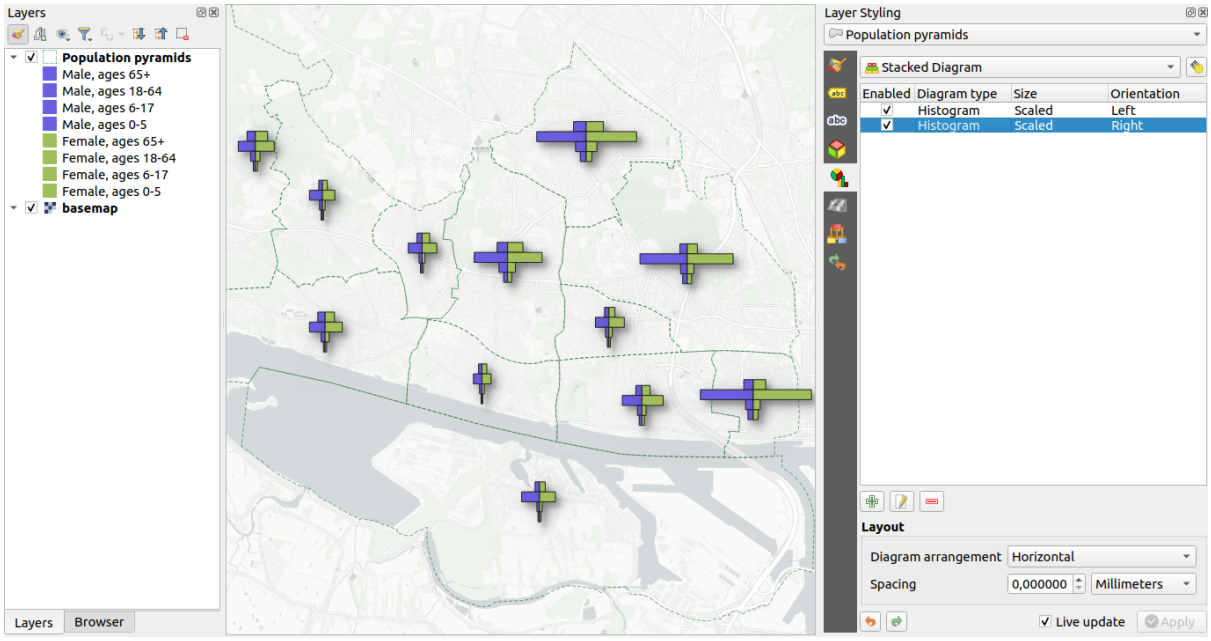
From the *Legend* tab, you can choose to display items of the diagram in the *Layers panel*, and in the *print layout legend*, next to the layer symbology:

- check *Show legend entries for diagram attributes* to display in the legends the `Color` and `Legend` properties, as previously assigned in the *Attributes* tab;
- and, when a *scaled size* is being used for the diagrams, push the *Legend Entries for Diagram Size…* button to configure the diagram symbol aspect in the legends. This opens the *Data-defined Size Legend* dialog whose options are described in *Data-defined size legend*.

When set, the diagram legend items (attributes with color and diagram size) are also displayed in the print layout legend, next to the layer symbology.

Stacked Diagrams

Stacked diagrams allow users to create complex diagrams like population pyramids, where two subdiagrams, namely histograms, are located side by side and displayed horizontally.



12.38 Fig. : Population pyramids built for each layer feature

Multi-temporal diagrams can also be constructed as stacked diagrams. The number of subdiagrams, as well as the spacing between them can be configured.

Moreover, subdiagrams can have different types (e.g., a pie chart alongside a histogram) and have their own independent settings like *Attributes*, *Rendering*, *Size*, *Options* and *Legend*.

Placement settings in a stacked diagram, as well as some visibility settings (located in the *Rendering* tab), are determined by the placement and visibility settings of the first subdiagram in the stack.

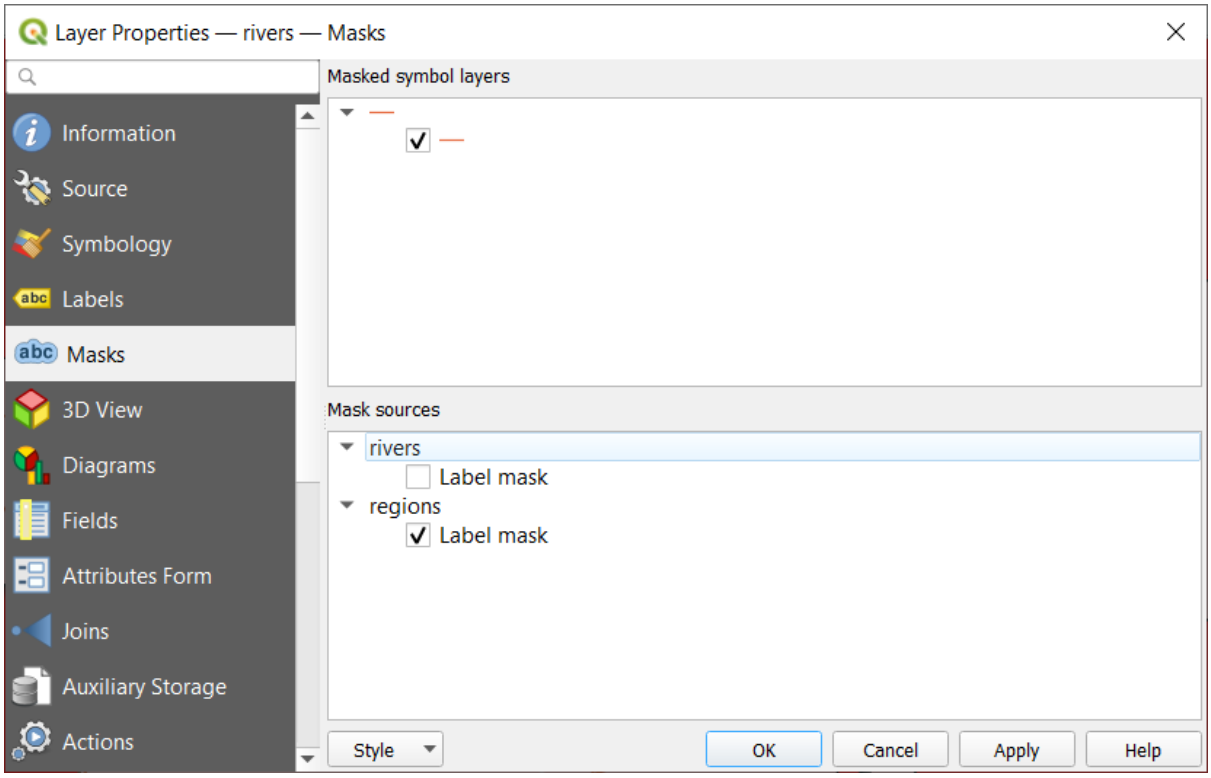
Finally, subdiagram ordering is given by the item ordering in the Stacked Diagram's list. The first subdiagram appears to the left in a horizontal stacked diagram, or in the upper part of a vertical one.

12.1.6 Masks Properties

The *Masks* tab helps you configure the current layer symbols overlay with other symbol layers or labels, from any layer. This is meant to improve the readability of symbols and labels whose colors are close and can be hard to decipher when overlapping; it adds a custom and transparent mask around the items to „hide“ parts of the symbol layers of the current layer.

To apply masks on the active layer, you first need to enable in the project either *mask symbol layers* or *mask labels*. Then, from the *Masks* tab, check:

- the *Masked symbol layers*: lists in a tree structure all the symbol layers of the current layer. There you can select the symbol layer item you would like to transparently „cut out“ when they overlap the selected mask sources
- the *Mask sources* tab: list all the mask labels and mask symbol layers defined in the project. Select the items that would generate the mask over the selected masked symbol layers



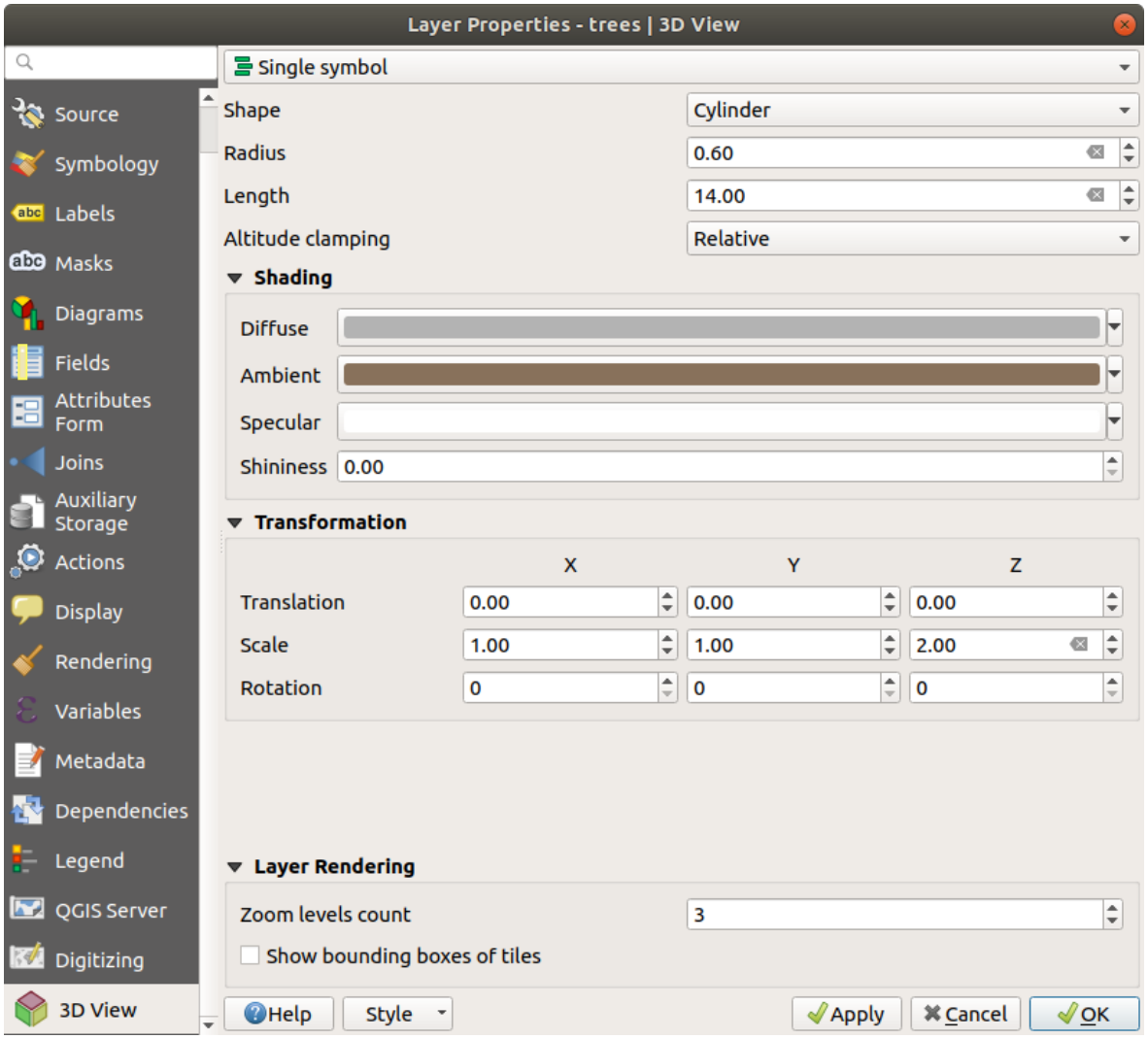
12.39 Fig. : Layer properties - Masks tab

12.1.7 3D View Properties

The *3D View* tab provides settings for vector layers that should be depicted in the *3D Map view* tool.

To display a layer in 3D, select from the combobox at the top of the tab, either:

- *Single symbol*: features are rendered using a common 3D symbol whose properties can be *data-defined* or not. Read details on *setting a 3D symbol* for each layer geometry type.
- *Rule-based*: multiple symbol configurations can be defined and applied selectively based on expression filters and scale range. More details on how-to at *Rule-based rendering*.



12.40 Fig. : 3D properties of a point layer

Dėmesio

Prefer the *Elevation* tab for symbol elevation and terrain settings

Features' elevation and altitude related properties (*Altitude clamping*, *Altitude binding*, *Extrusion* or *Height*) in the *3D View* tab inherit their default values from the layer's *Elevation properties* and should preferably be set from within the *Elevation* tab.

For better performance, data from vector layers are loaded in the background, using multithreading, and rendered in tiles whose size can be controlled from the *Layer rendering* section of the tab:

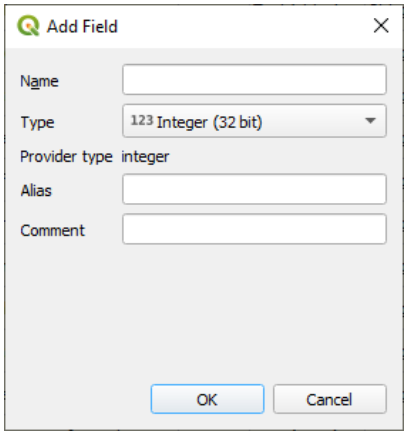
- *Zoom levels count*: determines how deep the quadtree will be. For example, one zoom level means there will be a single tile for the whole layer. Three zoom levels means there will be 16 tiles at the leaf level (every extra zoom level multiplies that by 4). The default is `3` and the maximum is `8`.
- *Show bounding boxes of tiles*: especially useful if there are issues with tiles not showing up when they should.

12.1.8 Fields Properties

The *Fields* tab provides information on fields related to the layer and helps you organize them.

The layer can be made *editable* using the Toggle editing mode. At this moment, you can modify its structure using the New field and Delete field buttons.

When creating New field, the *Comment* option is available only for data sources that allow editing comments (See *Duomenų bazių įrašai* for more details). You can also set aliases within *Add Field* dialog, for supported OGR formats (GeoPackage and ESRI File Geodatabase).



12.41 Fig. : Add Field Dialog

You can also rename fields by double-clicking its name. This is only supported for data providers like PostgreSQL, Oracle, Memory layer and some GDAL layers depending on the GDAL version.

If set in the underlying data source or in the *forms properties*, the field's alias is also displayed. An alias is a human readable field name you can use in the feature form or the attribute table. Aliases are saved in the project file.

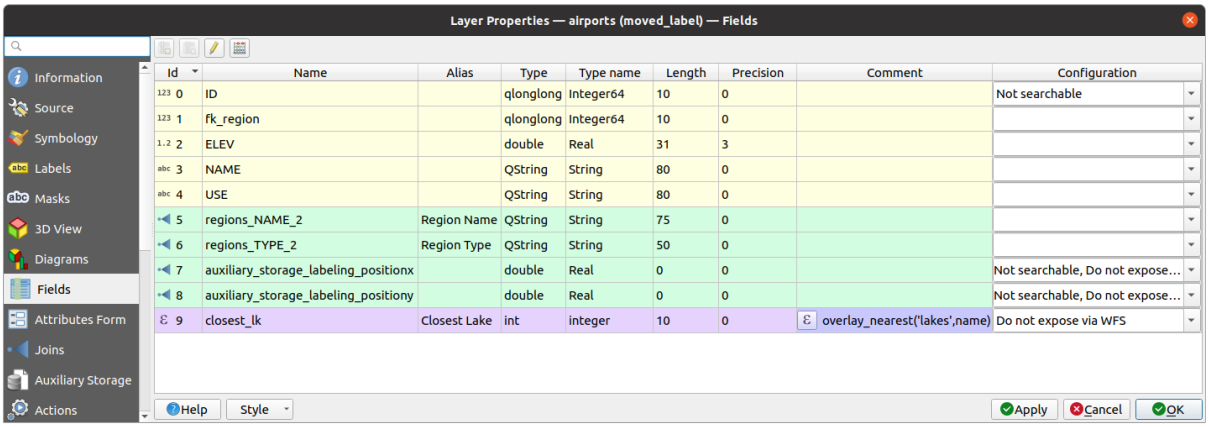
Other than the fields contained in the dataset, *virtual fields* and *Auxiliary Storage* included, the *Fields* tab also lists fields from any *joined layers*. Depending on the origin of the field, a different background color is applied to it.

For each listed field, the dialog also lists read-only characteristics such as its *Type*, *Type name*, *Length* and *Precision*`.

Depending on the data provider, you can associate a comment with a field, for example at its creation. This information is retrieved and shown in the *Comment* column and is later displayed when hovering over the field label in a feature form.

Under the *Configuration* column, you can set how the field should behave in certain circumstances:

- `Not searchable`: check this option if you do not want this field to be queried by the *search locator bar*
- `Do not expose via WMS`: check this option if you do not want to display this field if the layer is served as WMS from QGIS server
- `Do not expose via WFS`: check this option if you do not want to display this field if the layer is served as WFS from QGIS server



12.42 Fig. : Fields properties tab

12.1.9 Attributes Form Properties

The *Attributes Form* tab helps you set up the form to display when creating new features, editing or querying existing one. This affects them in both *table and form views*. You can define:

- the look and the behavior (label, visibility, widget, constraints…) of each field in the current layer, the *joined layers* and the *relational ones*;
- the placement of enabled *actions*;
- the form's structure (custom or autogenerated);
- extra HTML, QML, text widgets to enhance the form;
- extra logic in Python to handle interaction with the form or field widgets.

At the top right of the dialog, you can set whether the form is opened by default when creating new features. This can be configured per layer or globally with the *Suppress attribute form pop-up after feature creation* option in the *Settings* ► *Options* ► *Digitizing* menu.

Customizing a form for your data

By default, when you click on a feature with the Identify Features tool or switch the attribute table to the *form view* mode, QGIS displays a basic form with predefined widgets (generally spinboxes and textboxes — each field is represented on a dedicated row by its label next to the widget). If *relations* are set on the layer, fields from the referencing layers are shown in an embedded frame at the bottom of the form, following the same basic structure.

This rendering is the result of the default `Autogenerate` value of the *Attribute editor layout* setting in the *Layer properties* ► *Attributes Form* tab. This property holds three different values:

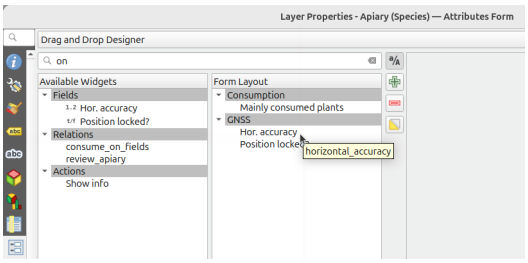
- `Autogenerate`: keeps the basic structure of „one row - one field“ for the form but allows to customize each corresponding widget.
- `Drag-and-drop designer`: other than widget customization, the form structure can be made more complex eg, with widgets embedded in groups and tabs.
- `Provide ui file`: allows to use a Qt designer file, hence a potentially more complex and fully featured template, as feature form.

Just below the *Attribute editor layout* setting, a search box allows you to filter fields, relations, containers and other available widgets.

For items that support aliases, like fields or relations, the filter targets both names and aliases.

The Show field aliases instead of names button allows you to toggle field aliases and field names.

When showing aliases, fields without an alias set will have their names displayed in light gray to help identify them easily.



12.43 Fig. : Search fields using aliases and names in Attributes Form page

The autogenerated form

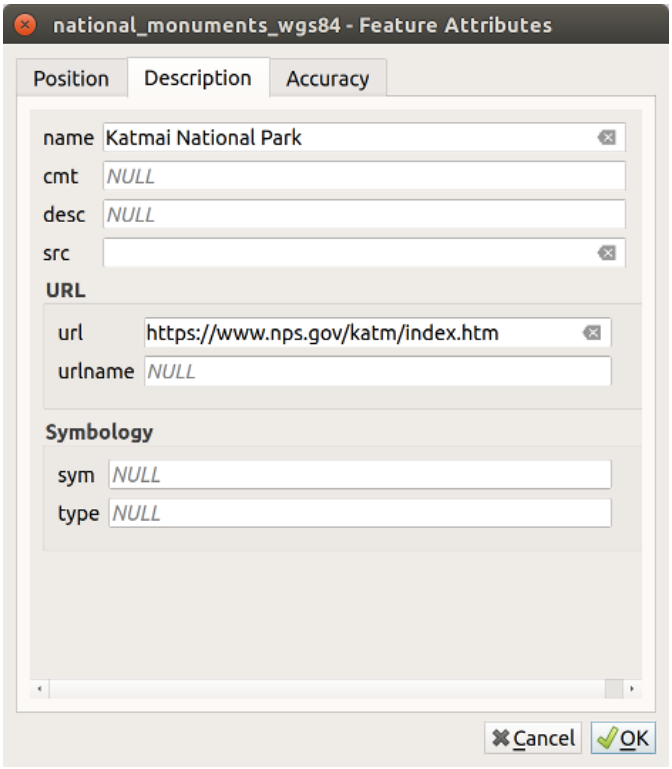
When the `Autogenerate` option is on, the *Available widgets* panel displays lists of fields (from the current layer, the join and relation layers) and the enabled actions that would be shown in the form. Select a field and you can configure its appearance and behavior in the right panel:

- adding *custom label and automated checks* to the field;
- setting a *particular widget* to use.

In this editor mode, only the properties can be viewed for the actions. Moreover, they will be displayed through an *Actions* drop-down menu in the feature form, and following the *Show in attribute table* property in other contexts.

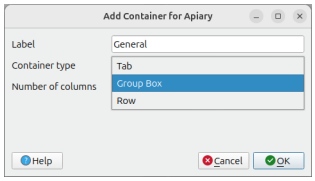
The drag and drop designer

The drag and drop designer allows you to create a form with several containers (tabs or groups) to present the attribute fields or other widgets that are not directly linked to a particular field (like the HTML/QML widgets or the *actions* defined for the layer), as shown for example in Fig. 12.44.



12.44 Fig. : Resulting built-in form with tabs and named groups

1. Choose `Drag and drop designer` from the *Select attribute layout editor* combobox. This enables the *Form Layout* panel next to the *Available widgets* panel, filled with existing fields. The selected field displays its *properties* in a third panel.
2. Select fields you do not want to use in your *Form Layout* panel and hit the button to remove them. You can also toggle the selection with the Invert selection button.
3. Drag and drop fields from the first panel to the *Form Layout* one to re-add them. The same field can be added multiple times.
4. Drag and drop fields within the *Form Layout* panel to reorder their position.
5. Add containers to associate fields that belong to the same category and better structure the form.
    1. The first step is to use the Add a new tab or group to the form layout icon. Fields and other groups will be displayed in it.
    2. Then set the properties of the container, i.e.:
        - the *Label*: the title that will be used for the container
        - the *Container Type*: it can be a *Tab*, *Group box in container* (a collapsible group box inside a tab or another group) or a *Row* (a container type that allows you to arrange your widgets in a horizontal row, automatically determining the number of columns based on the number of widgets),
        - the *Within*: this optional feature allows you to select an existing container in which the new container (*Group box in container* or *Row*) will be embedded.
        - and the *Number of columns* the embedded fields should be distributed over



12.45 Fig. : Dialog to create containers with the **Attribute editor layout**

These, and other properties can later be updated by selecting the item and, from the third panel:

- hide or show the container's label
- rename the container
- set the number of columns
- enter an expression to control the container's visibility. The expression will be re-evaluated every time values in the form change, and the tab or group box shown/hidden accordingly
- *Show as Group Box*: converts a tab into a top-level group box and vice versa
- in case of a group box, set whether at the form opening, it should show as *Collapsed* for every features, or only for those matching an expression (*Control collapsed by expression*).
- configure the *Style* look of the container, with custom *Background color*, label color and font properties

3. You can create and embed as many containers as you want by pressing again the Add a new tab or group to the form layout icon.

6. The next step is to assign the relevant fields to each container, by simple drag and drop. Group boxes and tabs can also be moved the same way.
7. *Customize the widget* of the fields in use
8. In case the layer is involved in a *one or many to many relation*, drag-and-drop the relation name from the *Available Widgets* panel to the *Form Layout* panel. The associated layer attribute form will be embedded at the chosen place in the current layer's form. As for the other items, select the relation label to configure some properties:
   - hide or show the relation label
   - show the link button
   - show the unlink button
9. In case the layer has one or more *actions* enabled for `Layer` or `Feature` scope, the actions will be listed under *Actions* and you can drag and drop them as with the other fields. The associated action will be embedded at the chosen place in the current layer's form.
10. Further customize the form by adding one or more widgets from *Other Widgets* (see *Other Widgets*)
11. Apply the layer's properties dialog
12. Open a feature attribute form (eg, using the Identify features tool) and it should display the new form.

Other Widgets

The drag and drop designer offers a number of widgets that are not connected to a particular field. They can be used to enhance the appearance of the form or to display dynamically calculated values.

- *HTML Widget*: embeds an HTML page, the HTML source may contain the result of dynamically calculated expressions.

  HTML widgets can be used for example to display images stored as BLOB in a field (let's call it `photo`):

  1. In the `Drag-and-drop designer` mode, add a *HTML Widget* to your *Form Layout*.
  2. Double-click on the *HTML Widget* to configure it.
  3. Change the default *Title* or hide it.
  4. Press the ε button and enter the following QGIS expression:

     ```
     '<img src= "' || to_base64("photo") || '">'
     ```

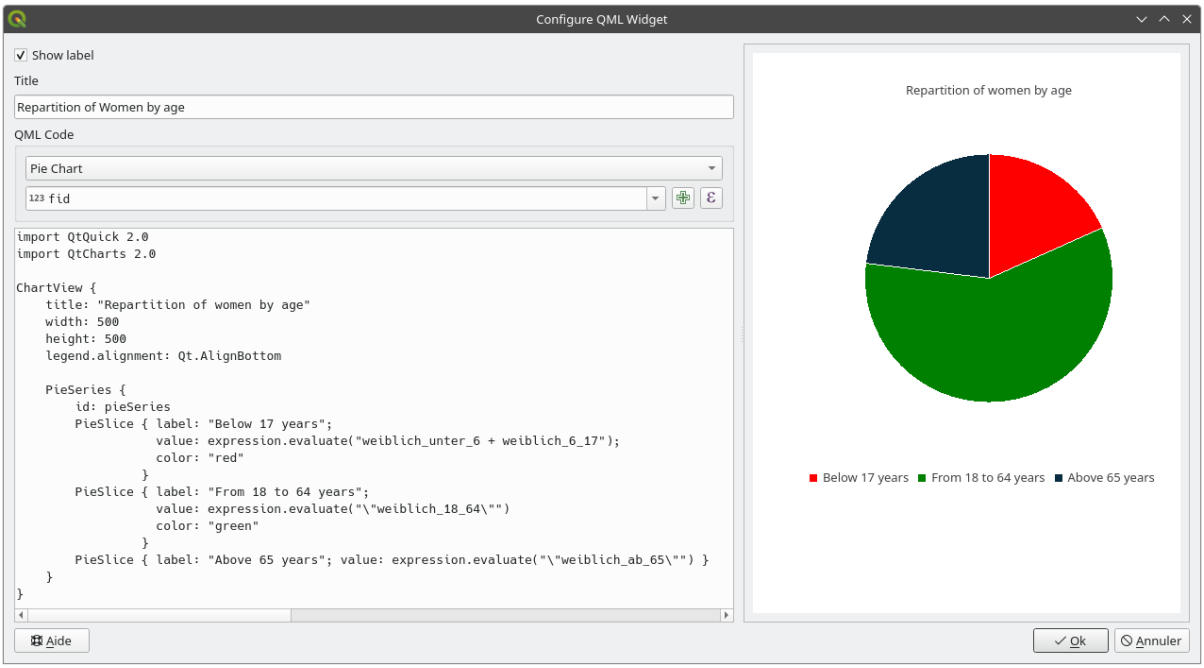
     Ensure that you replace *photo* with your own BLOB field name. The above expression creates a string with HTML image tag in which the BLOB file is encoded.
  5. Apply the dialog and then press the ✚ button.
  6. QGIS automatically applies HTML formatting and functions to evaluate your expression, resulting in following code:

     ```
     <script>document.write(expression.evaluate("'<img src=\"data:image/png;
     ↪base64,' || to_base64(\"photo\") || '\">'"));</script>
     ```

     A preview of your image is displayed on the right.

- *QML Widget*: embeds a Qt QML document, displaying graphical elements in the attribute form. Beside the custom *Title* of the added widget and whether it should be shown or not, you can select from predefined *QML code* elements:
  - *Free text...*: allows you to write from scratch or paste an existing code
  - *Rectangle*: provides a minimal code for displaying a rectangle
  - *Bar chart*: provides a minimal code for displaying a bar chart
  - *Pie chart*: provides a minimal code for displaying a pie chart

  You can extend the code with QML syntax, use layer fields or QGIS expressions that are dynamically calculated.



12.46 Fig. : Setting a QML graph to display in attribute form

- *Text Widget*: displays a text widget which supports basic HTML markup and may contain the result of dynamically calculated expressions.
- *Spacer Widget*: inserts an empty transparent rectangle, increasing the vertical distance between two widgets.

Patarimas

Display Dynamic Content

The widgets mentioned above (except the *Spacer Widget*) support expressions that can be used to display content that dynamically changes whenever another field in the form changes. This can be achieved using the `current_value('field_name')` function in the expression to examine the value of another field.

Using custom ui-file

The `Provide ui-file` option allows you to use complex dialogs made with Qt-Designer. Using a UI-file allows a great deal of freedom in creating a dialog. Note that, in order to link the graphical objects (textbox, combobox…) to the layer's fields, you need to give them the same name.

Use the *Edit UI* to define the path to the file to use.

UI-files can also be hosted on a remote server. In this case, you provide the URL of the form instead of the file path in *Edit UI*.

You'll find some example in the Creating a new form lesson of the QGIS-training-manual-index-reference. For more advanced information, see https://woostuff.wordpress.com/2011/09/05/qgis-tips-custom-feature-forms-with-python-logic/.

Enhance your form with custom functions

QGIS forms can have a Python function that is called when the dialog is opened. Use this function to add extra logic to your dialogs. The form code can be specified in three different ways:

- `load from the environment`: use a function, for example in `startup.py` or from an installed plugin
- `load from an external file`: a file chooser will let you select a Python file from your filesystem or enter a URL for a remote file.
- `provide code in this dialog`: a Python editor will appear where you can directly type the function to use.

In all cases you must enter the name of the function that will be called (`open` in the example below).

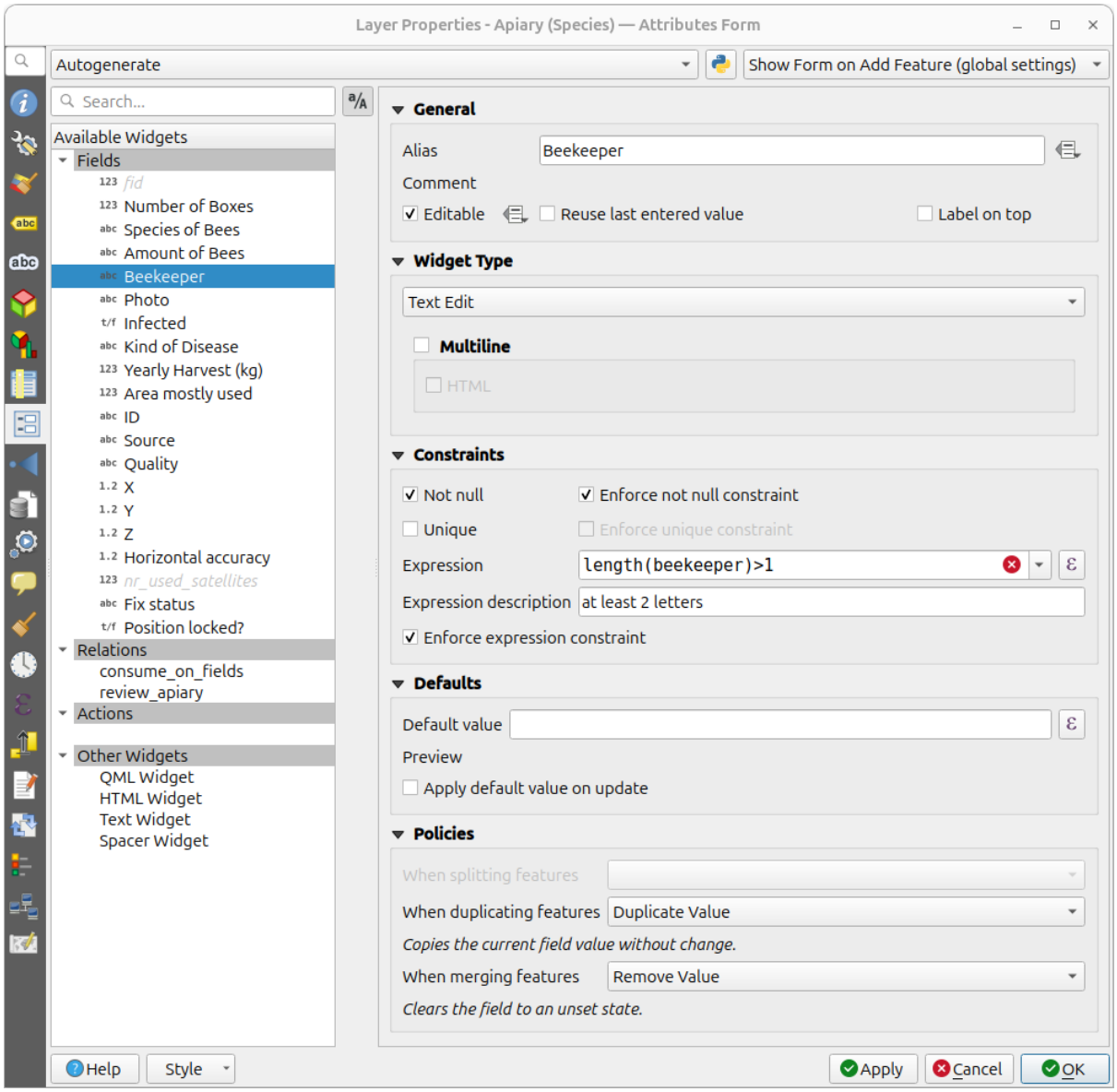
An example is (in module MyForms.py):

```
def open(dialog,layer,feature):
    geom = feature.geometry()
    control = dialog.findChild(QWidget,"My line edit")
```

Reference in Python Init Function like so: `open`

Configure the field behavior

The main part of the *Attributes Form* tab helps you set the type of widget used to fill or display values of the field, in the attribute table or the feature form: you can define how user interacts with each field and the values or range of values that are allowed to be added to each.



12.47 Fig. : Dialog to select an edit widget for an attribute column

Patarimas

Copy and paste widget configuration

The field configuration can be copied to the clipboard and then pasted to another field via right click on a field in the *Available widgets* panel.

This comes handy to save time and to make sure that common settings are correctly applied to several related fields, e.g., when configuring latitude and longitude field widgets.

Copy and paste of widget configuration can be done between fields of the same layer, between fields from different layers in the QGIS project, or between fields from layers in different QGIS instances.

Note that neither aliases nor comments are copied, since those are specific to each particular field.

Common settings

Regardless the type of widget applied to the field, there are some common properties you can set to control whether and how a field can be edited.

Widget display

Only available for the *Drag and drop* designer mode, this group helps you configure the look of the widget assigned to the field:

- *Show label*: indicates whether the field name should be displayed in the form
- *Override label color*: applies specific color to the field's label
- *Override label font*: applies specific font properties (bold, italic, underline, strikethrough, or font family) to the field's label
- *Size*: allows to control how widgets will relatively resize when resizing an attribute form.
  - *Horizontal stretch*: sets a higher horizontal value for widgets that need more horizontal space.
  - *Vertical stretch*: determines how widgets resize vertically when the form is resized.

General options

- *Alias*: a human readable name to use for fields. The alias will be displayed in the feature form, the attribute table, or in the *Identify results* panel. It can also be used as field name replacement in the *expression builder*, easing expressions understanding and reviews. Aliases are saved in project file.
- *Comment*: displays the field's comment as shown in the *Fields* tab, in a read-only state. This information is shown as tooltip when hovering over the field label in a feature form.
- *Editable*: uncheck this option to set the field read-only (not manually modifiable) even when the layer is in edit mode. Note that checking this setting doesn't override any edit limitation from the provider. This option can be controlled by a data-defined property thanks to the Data-defined override button.
- *Reuse last entered value*: remembers the last value entered in this field and uses it as default for the next feature being edited in the layer.
- *Label on top*: places the field name above or beside the widget in the feature form.

Constraints

You can constrain the value to insert in the field. This constraint can be:

- *Not null*: requires the user to provide a value;
- *Unique*: guarantee the inserted value to be unique throughout the field;
- based on a custom *expression*: e.g. `not regexp_match(col0,'[^A-Za-z]')` will ensure that the value of the field *col0* has only alphabet letters. A short description can be added to help you remember the constraint.

Whenever a value is added or edited in a field, it's submitted to the existing constraints and:

- if it meets all the requirements, a green check is shown beside the field in the form;
- if it does not meet all the requirements, then the field is colored in yellow or orange and a corresponding cross is displayed next to the widget. You can hover over the cross to remind which constraints are applied to the field and fix the value:

  - A yellow cross appears when the unmet constraint is an unenforced one (`soft constraint`) and it does not prevent you to save the changes with the „wrong“ values;
  - An orange cross can not be ignored and does not allow you to save your modifications until they meet the constraints. It appears when the *Enforce constraint* option is checked (`hard constraint`).

Default values

- *Default value*: for new features, automatically populates by default the field with a predefined value or an *expression-based one*. For example, you can:
  - use `$x`, `$length`, `$area` to automatically populate a field with the feature's X coordinate, length, area or any geometric information at its creation;
  - increment a field by 1 for each new feature using `maximum("field")+1`;
  - save the feature creation datetime using `now()`;
  - use *variables* in expressions, making it easier to e.g. insert the operator name (`@user_full_name`), the project file path (`@project_path`), …

  A preview of the resulting default value is displayed at the bottom of the widget.

  Pastaba

  The `Default value` option is not aware of the values in any other field of the feature being created so it won't be possible to use an expression combining any of those values i.e using an expression like `concat(field1, field2)` may not work.

- *Apply default value on update*: whenever the feature attribute or geometry is changed, the default value is recalculated. This could be handy to save values like last user that modifies data, last time it was changed…

Policies

Field policies determine how values are assigned to fields during various editing operations:

Split and Duplicate Policies

These policies apply when *Splitting features* or *Duplicating features*:

- *Duplicate Values*: Keeps the existing value of the field for the new features.
- *Use Default Value*: Resets the field by recalculating its *default value*. If no default value clause exists, the existing value is kept for the new features.
- *Remove Value*: Clears the field to an unset state.
- *Use Ratio Geometries*: Recalculates the field value for all split portions by multiplying the existing value by ratio of the split parts lengths or areas.

Merge Policies

These policies determine initial values when *Merging features*:

- *Remove Value*: Clears the field to an unset state (data provider may populate default value).
- *Use Default Value*: Uses the default field value set in QGIS. (expressions use a dummy feature with the merged geometry; references to other fields won't work).
- *Use Sum* (numeric fields only): Sums values from all merged features.
- *Use Minimum Value* (numeric fields only): Uses smallest value from the selected features.
- *Use Maximum Value* (numeric fields only): Uses largest value from the selected features.

- *Average Weighted by Geometry* (numeric fields only): New values are computed as the weighted average of the source values.
- *Use Largest Feature*: Uses value from feature with the largest geometry.
- *Set to NULL*: Sets attribute to NULL.

Edit widgets

Based on the field type, QGIS automatically determines and assigns a default widget type to it. You can then replace the widget with any other compatible with the field type. The available widgets are:

- **Binary (BLOB)**: Available only for binary fields, it displays by default a label with the size of the embedded data, if not empty. A drop-down button next to the label allows to:
  - *Embed file*, replacing or filling the field
  - *Clear contents*, removing any data in the field
  - *Save contents to file*, exporting the data as a file on disk

  It is also possible to preview the embedded binary file in the field, if combined in a drag-and-drop form with e.g. a *QML or HTML widget*.
- **Checkbox**: Displays a checkbox whose state defines the value to insert. Internally, the widget stores a value (e.g. 1 for checked, 0 for unchecked) while representing it visually as a toggled box. You can configure the stored values for the checked and unchecked states in the representation. It is also possible to display the checkbox state as a plain text using „True“/“False“ labels. These settings influence how the values are shown in classification renderers.
- **Classification**: Only available when a *categorized symbology* is applied to the layer, displays a combo box with the values of the classes.
- **Color**: Displays a *color widget* allowing to select a color; the color value is stored as a html notation in the attribute table.
- **Date/Time**: Displays a line field which can open a calendar widget to enter a date, a time or both. Column type must be text. You can select a custom format, pop-up a calendar, etc.
- **Enumeration**: Opens a combo box with predefined values fetched from the database. This is currently only supported by the PostgreSQL provider, for fields of `enum` type.
- **Attachment**: Uses a „Open file“ dialog to store file path in a relative or absolute mode. It can be used to display the document path as a hyperlink or render the document within a dedicated widget in the form. Supported document types are image, web page, audio and video, and the supported file formats depend on the Operating System. You can also configure an *external storage system* to fetch/store resources.

  > **Patarimas**
  >
  > **Relative Path in Attachment widget**
  >
  > If the path which is selected with the file browser is located in the same directory as the `.qgs` project file or below, paths are converted to relative paths. This increases portability of a `.qgs` project with multimedia information attached.

- **Geometry**: Compatible with the *Geometry* field type. For example, PostgreSQL and Memory layers can contain multiple geometry fields. QGIS uses the first geometry column as the layer geometry. Applying the Geometry widget to additional geometry fields allows lossless use of the data in the attribute form:
  - Copy in the formats WKT (Well-known text representation, e.g. `POINT (30 10)`) or GeoJSON
  - Paste
  - Clear

- **Hidden**: A hidden attribute column is invisible. The user is not able to see its contents.
- **Key/Value**: Displays a two-columns table to store sets of key/value pairs within a single field. This is currently supported by the PostgreSQL provider, for fields of `hstore` type.
- **JSON View**: Displays JSON data in a syntax highlighted text edit or in tree view. This widget is currently read only. Several options are available to change how the data is displayed. ‚Default view' specify if the widget should appear in Text or Tree mode. ‚Format JSON' has three options which are related to the tree view only:
  - Indented: Display data in a human readable form with newlines and four space characters for indentation.
  - Compact: Display data in a one-line size optimized string without newlines or spaces.
  - Disabled: Display data as it comes from the provider.
- **List**: Displays a single column table to add different values within a single field. This is currently supported by the PostgreSQL provider, for fields of `array` type.
- **Range**: Allows you to set numeric values from a specific range. The edit widget can be either a slider or a spin box.

  > **Pastaba**
  >
  > Some layers, such as GeoPackage or ESRI File Geodatabase, with predefined **range Field Domains** will be automatically recognized by QGIS and assigned a **Range** widget for the relevant fields. The widget will be prefilled with the minimum and maximum values specified in the domain.

- **Relation Reference**: This is the default widget assigned to the referencing field (i.e., the foreign key in the child layer) when a *relation* is set. It provides direct access to the parent feature's form which in turn embeds the list and form of its children. The number of entries in the widget can be limited for efficiency, and if limit is not set, all entries will be loaded. The field defined in the display expression is used as representable value in classification renderers.
- **Text Edit** (default): This opens a text edit field that allows simple text or multiple lines to be used. If you choose multiple lines you can also choose html content.
- **Unique Values**: You can select one of the values already used in the attribute table. If ‚Editable' is activated, a line edit is shown with autocompletion support, otherwise a combo box is used.
- **Uuid Generator**: Generates a read-only UUID (Universally Unique Identifiers) field, if empty.
- **Value Map**: A combo box with predefined items. The value is stored in the attribute, the description is shown in the combo box. You can define values manually or load them from a layer or a CSV file. The description is used as the representable value, meaning it will be shown in classification renderers.

  > **Pastaba**
  >
  > Some layers, such as GeoPackage or ESRI File Geodatabase, with predefined **coded Field Domains** will be automatically recognized by QGIS and assigned a **Value Map** widget for the relevant fields.

- **Value Relation**: Offers values from a related table in a combobox. You can select layer, key column and value column. The key column is the stored value in the attribute table, while the value column is the label shown in the combobox. Value column is also the representable value that will be shown in the classification renderers. You may also choose to order the list of displayed values by the *Key* column (the default sort order), the *Value* column, or a specific *Field* of the layer. Enable the *Descending order* if you wish the values to be sorted in reverse order.

  Several options are available to change the standard behavior: allow NULL value, allow multiple selections and use of auto-completer. The forms will display either a drop-down list or a line edit field when completer checkbox is enabled.

If a layer that is stored in PostgreSQL, GeoPackage or SpatiaLite is configured to use a value relation widget, but the required layer is not already loaded into the project, QGIS will automatically search for the layer in the same database/connection.

12.1.10 Joins Properties

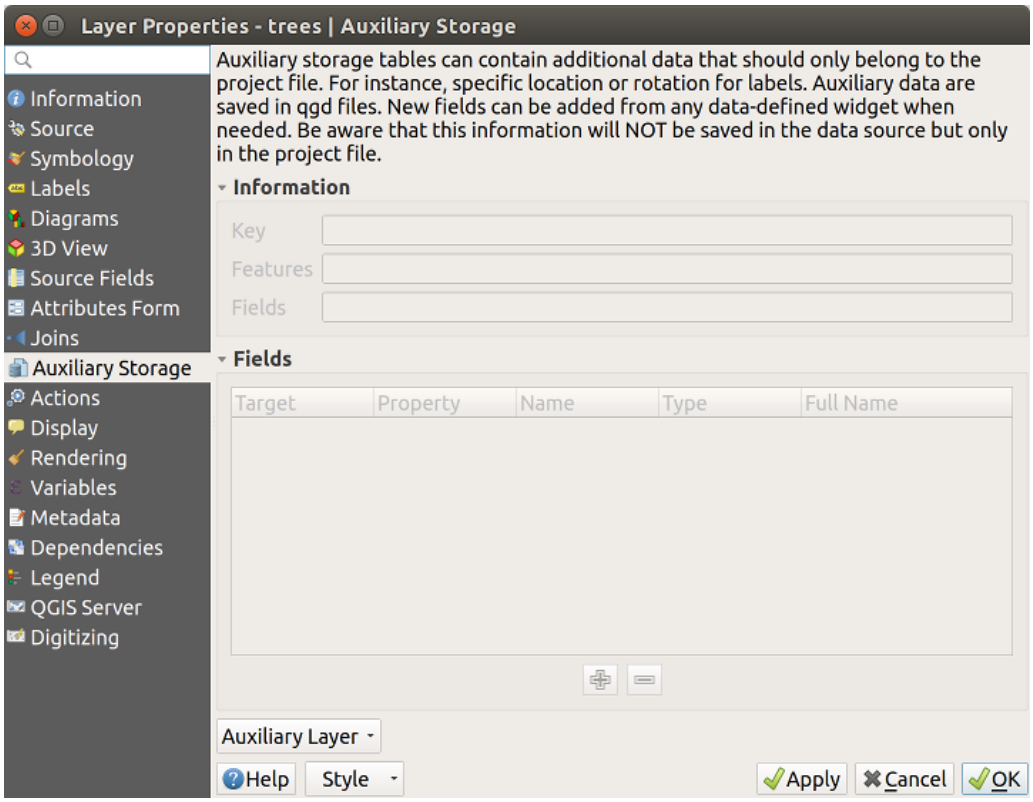
The *Joins* tab allows you to associate features of the current layer to features from another loaded vector layer (or table). The join is based on an attribute that is shared by the layers, in a one-to-one relationship. For more details on joins, please read *Dviejų sluoksnių geoobjektų sujungimas*.

12.1.11 Auxiliary Storage Properties

The regular way to customize styling and labeling is to use data-defined properties as described in *Nuo duomenų priklausančių permušimų nustatymai*. However, it may not be possible if the underlying data is read only. Moreover, configuring these data-defined properties may be very time consuming or not desirable! For example, if you want to fully use map tools coming with *The Label Toolbar*, then you need to add and configure more than 20 fields in your original data source (X and Y positions, rotation angle, font style, color and so on).

The Auxiliary Storage mechanism provides the solution to these limitations and awkward configurations. Auxiliary fields are a roundabout way to automatically manage and store these data-defined properties (labels, diagram, symbology…) in a SQLite database thanks to editable joins. This allows you to store properties for layers that aren't editable.

A tab is available in vector layer properties dialog to manage auxiliary storage:

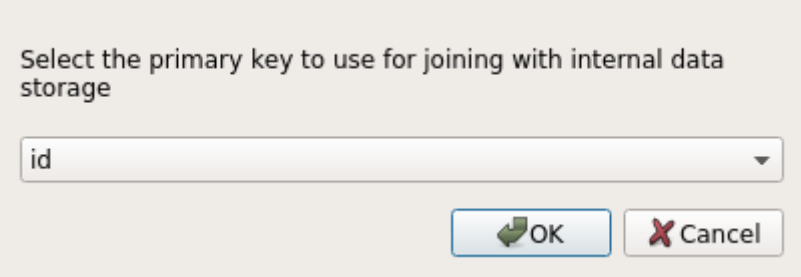


12.48 Fig. : Auxiliary Storage tab

Labeling

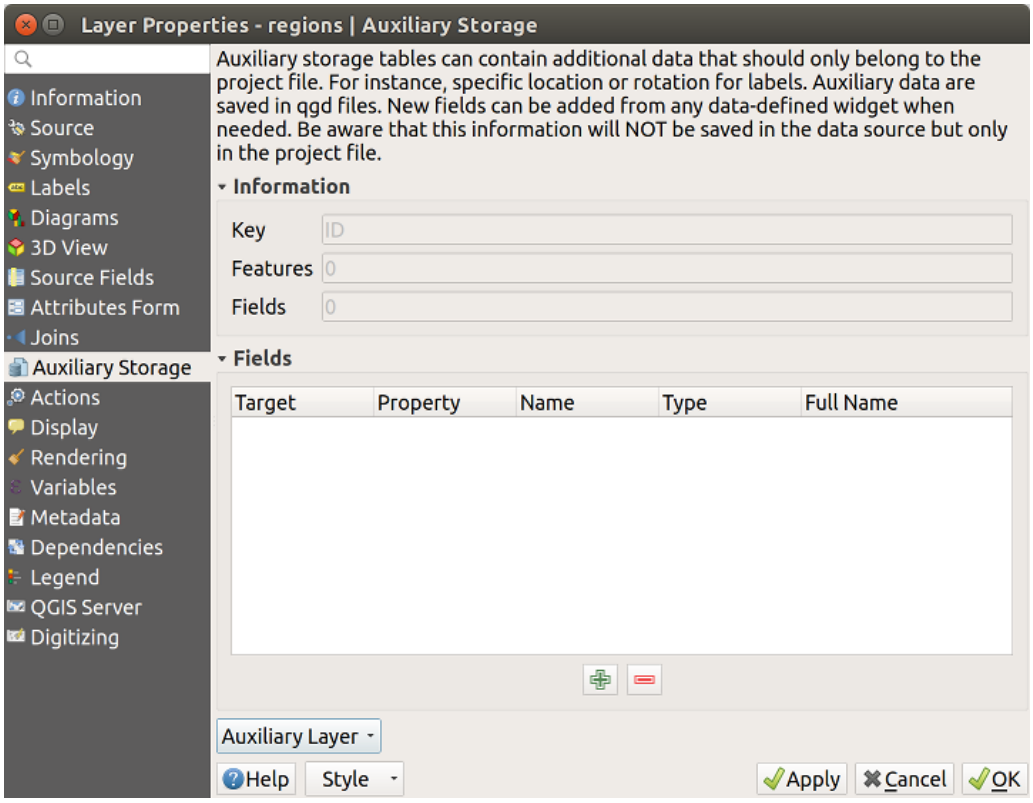
Considering that the data source may be customized thanks to data-defined properties without being editable, labeling map tools described in *The Label Toolbar* are always available as soon as labeling is activated.

Actually, the auxiliary storage system needs an auxiliary layer to store these properties in a SQLite database (see *Auxiliary storage database*). Its creation process is run the first time you click on the map while a labeling map tool is currently activated. Then, a window is displayed, allowing you to select the primary key to use for joining (to ensure that features are uniquely identified):



12.49 Fig. : Auxiliary Layer creation dialog

As soon as an auxiliary layer is configured for the current data source, you can retrieve its information in the tab:

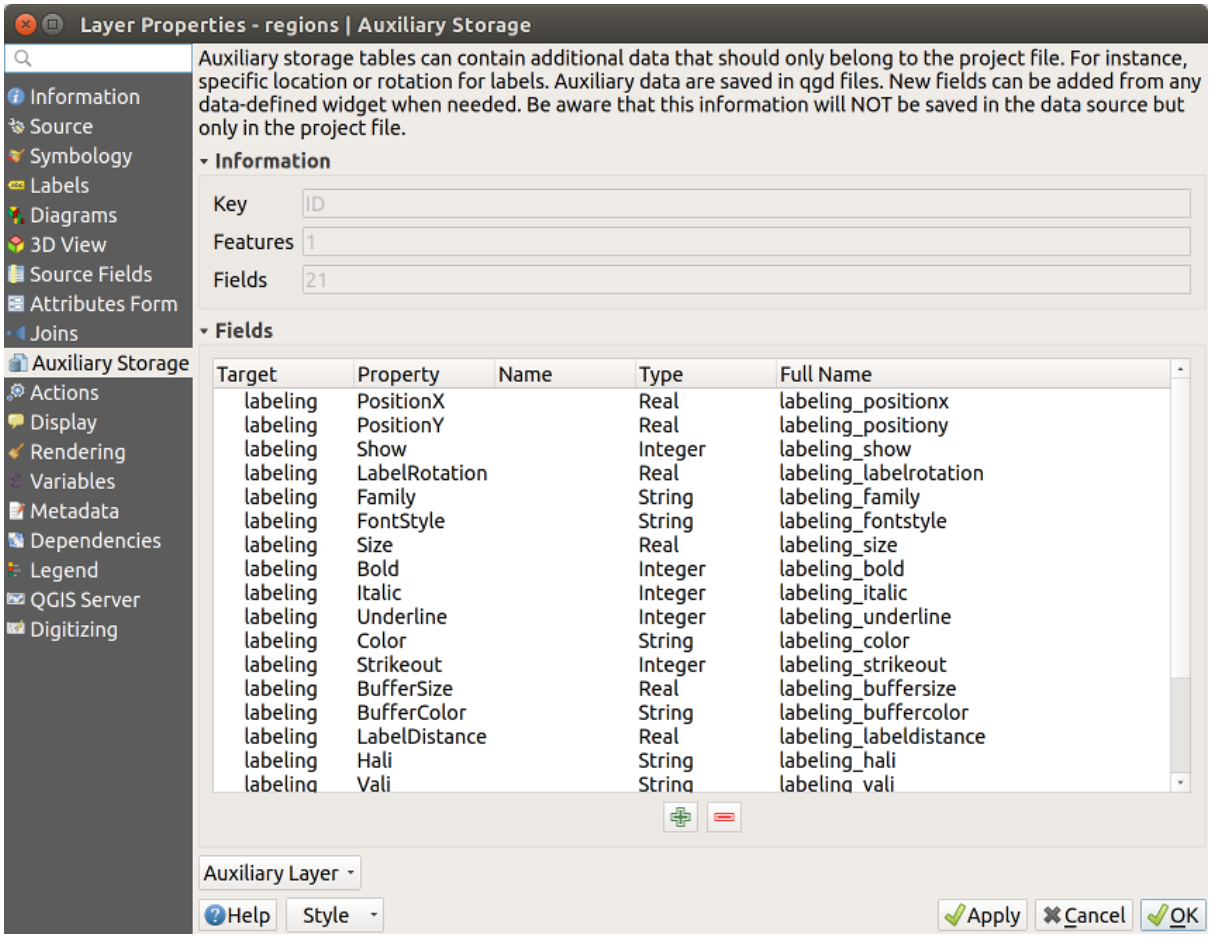


12.50 Fig. : Auxiliary Layer key

The auxiliary layer now has these characteristics:

- the primary key is `ID`,
- there are `0` features using an auxiliary field,
- there are `0` auxiliary fields.

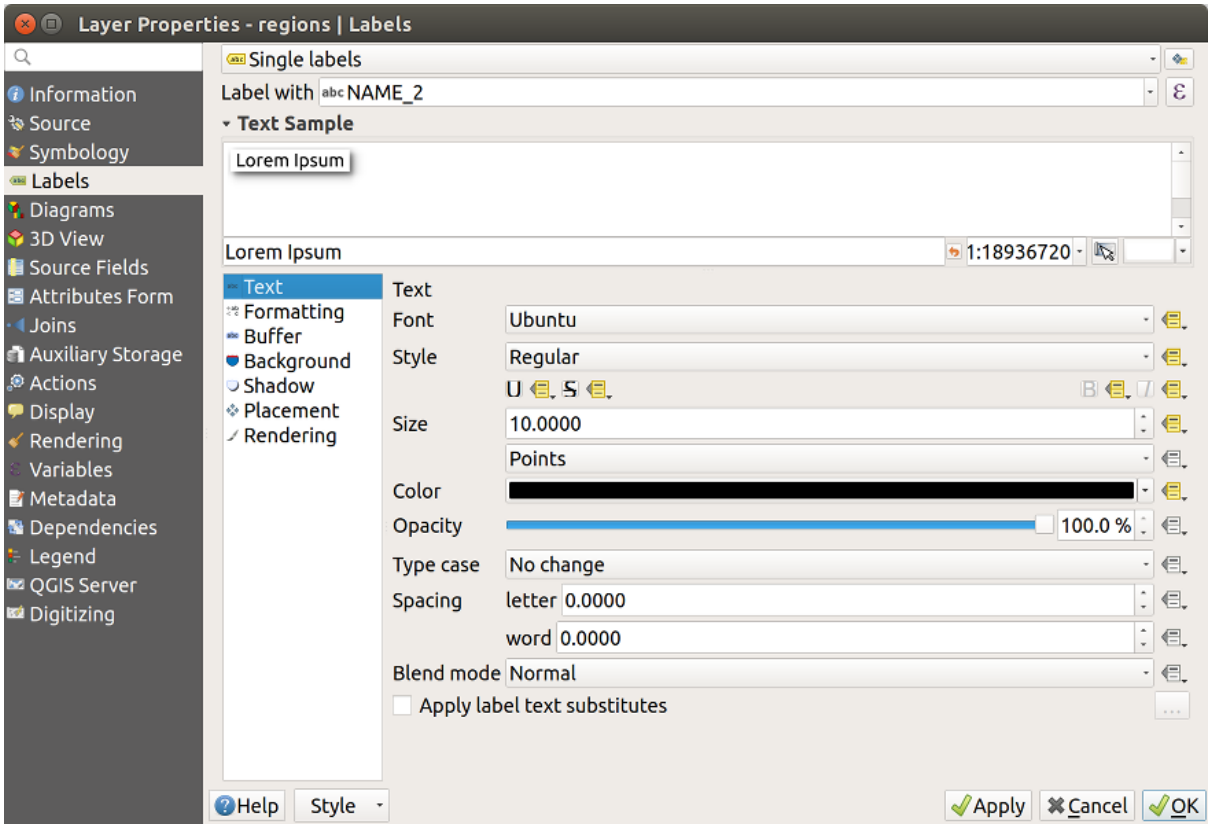
Now that the auxiliary layer is created, you can edit the layer labels. Click on a label while the Change Label map tool is activated, then you can update styling properties like sizes, colors, and so on. The corresponding data-defined properties are created and can be retrieved:



12.51 Fig. : Auxiliary Fields

As you can see in the figure above, `21` fields are automatically created and configured for labeling. For example, the `FontStyle` auxiliary field type is a `String` and is named `labeling_fontstyle` in the underlying SQLite database. There is also `1` feature which is currently using these auxiliary fields.

Notice that the icon is displayed in the *Labels* properties tab indicating that the data-defined override options are set correctly:

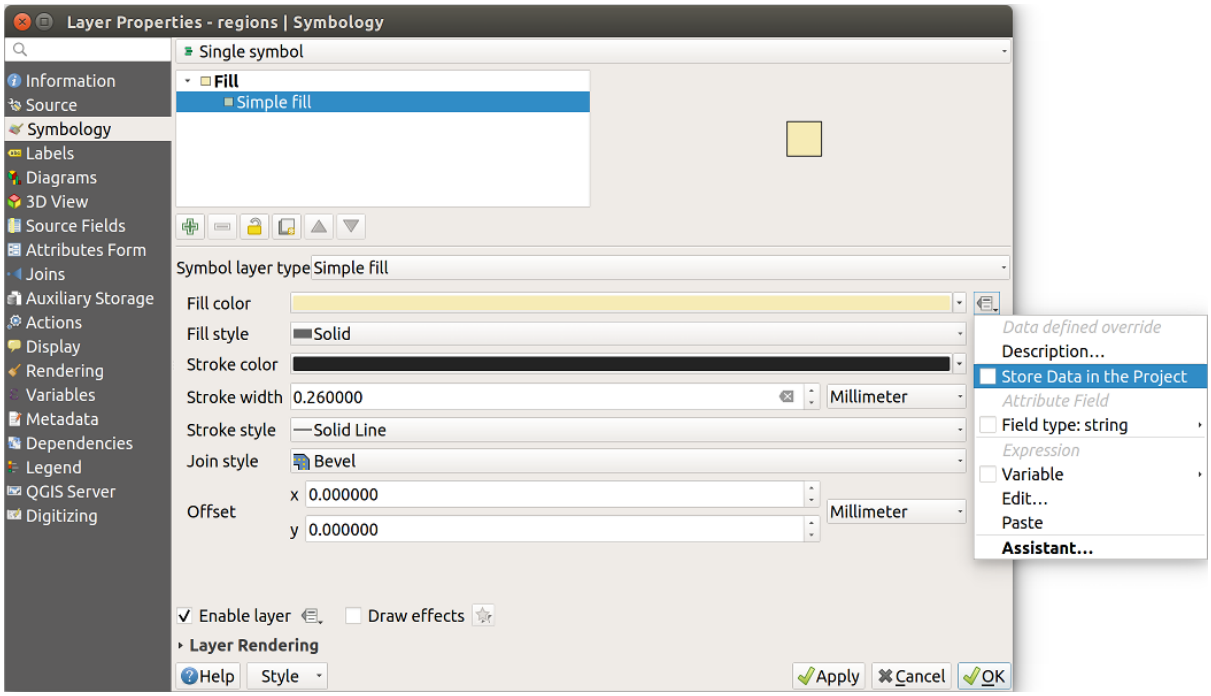


12.52 Fig. : Data-defined properties automatically created

Otherwise, there's another way to create an auxiliary field for a specific property thanks to the Data-defined override button. By clicking on *Store data in the project*, an auxiliary field is automatically created for the *Opacity* field. If you click on this button and the auxiliary layer is not created yet, a window (Fig. 12.49) is first displayed to select the primary key to use for joining.

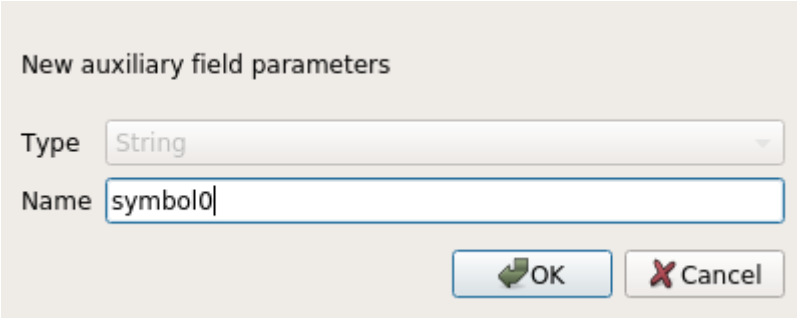
Simbologija

Like the method described above for customizing labels, auxiliary fields can also be used to stylize symbols and diagrams. To do this, click on Data-defined override and select *Store data in the project* for a specific property. For example, the *Fill color* field:



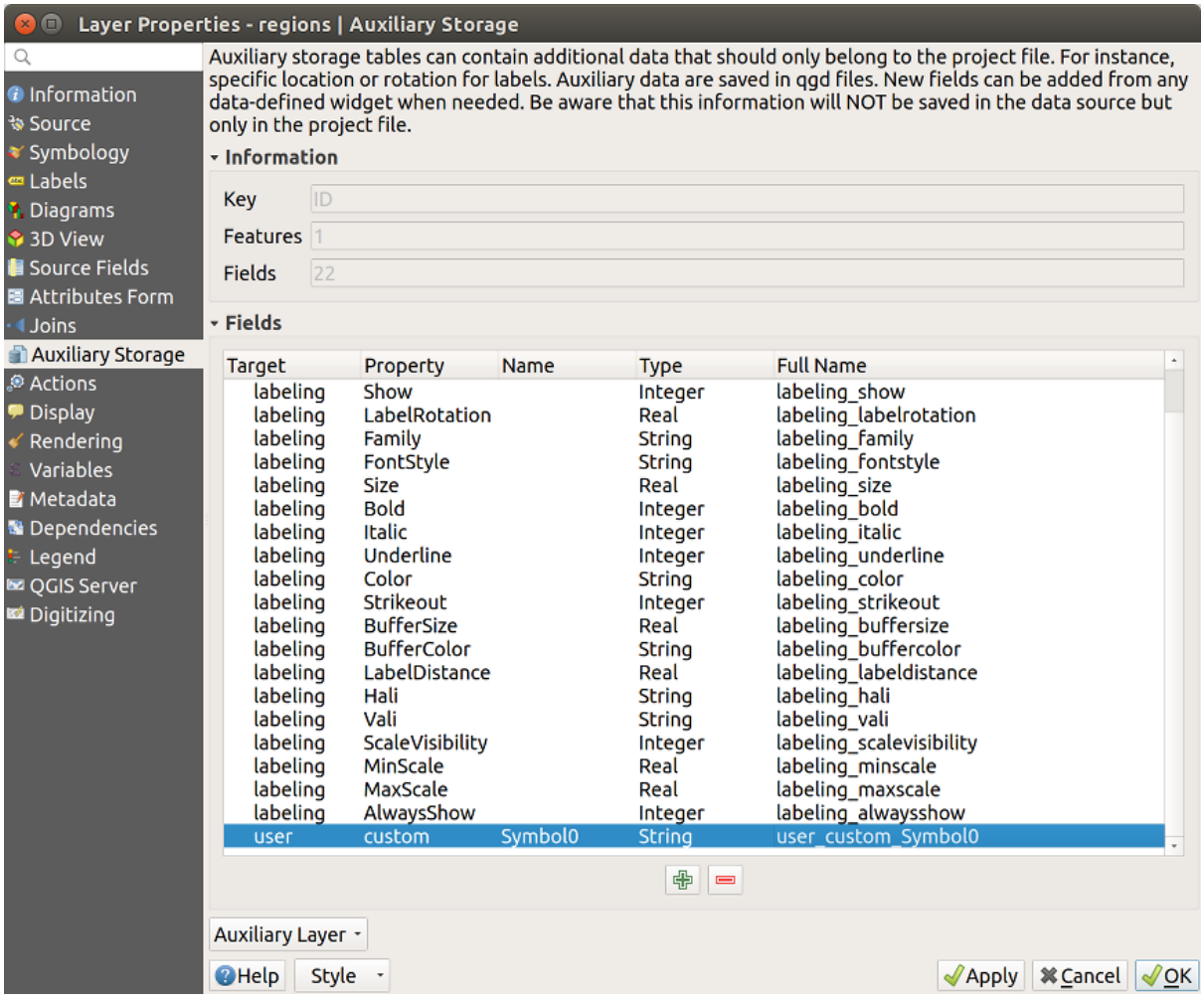
12.53 Fig. : Data-defined property menu for symbol

There are different attributes for each symbol (e.g. fill style, fill color, stroke color, etc…), so each auxiliary field representing an attribute requires a unique name to avoid conflicts. After selecting *Store data in the project*, a window opens and displays the *Type* of the field and prompts you to enter a unique name for the auxiliary field. For example, when creating a *Fill color* auxiliary field the following window opens:



12.54 Fig. : Name of the auxiliary field for a symbol

Once created, the auxiliary field can be retrieved in the auxiliary storage tab:



12.55 Fig. : Auxiliary field symbol

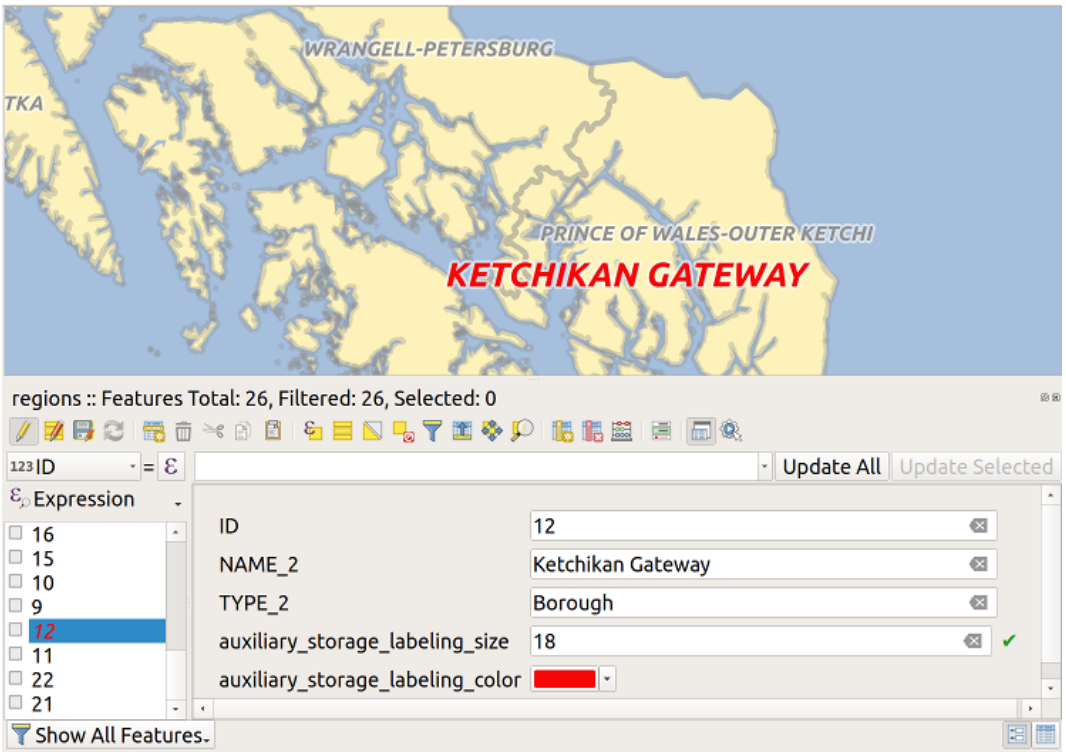
Attribute table and widgets

Auxiliary fields can be edited using the *attribute table*. However, not all auxiliary fields are initially visible in the attribute table.

Auxiliary fields representing attributes of a layer's symbology, labeling, appearance, or diagrams will appear automatically in the attribute table. The exception are attributes that can be modified using the *Label Toolbar* which are hidden by default. Auxiliary fields representing a `Color` have a widget **Color** set by default, otherwise auxiliary fields default to the **Text Edit** widget.

Auxiliary fields that represent attributes that can be modified using the *Label toolbar* are **Hidden** in the attribute table by default. To make a field visible, open the *Attribute Form properties tab* and change the value of an auxiliary field *Widget Type* from **Hidden** to another relevant value. For example, change the **auxiliary_storage_labeling_size** to **Text Edit** or change **auxiliary_storage_labeling_color** to the **Color** widget. Those fields will now be visible in the attribute table.

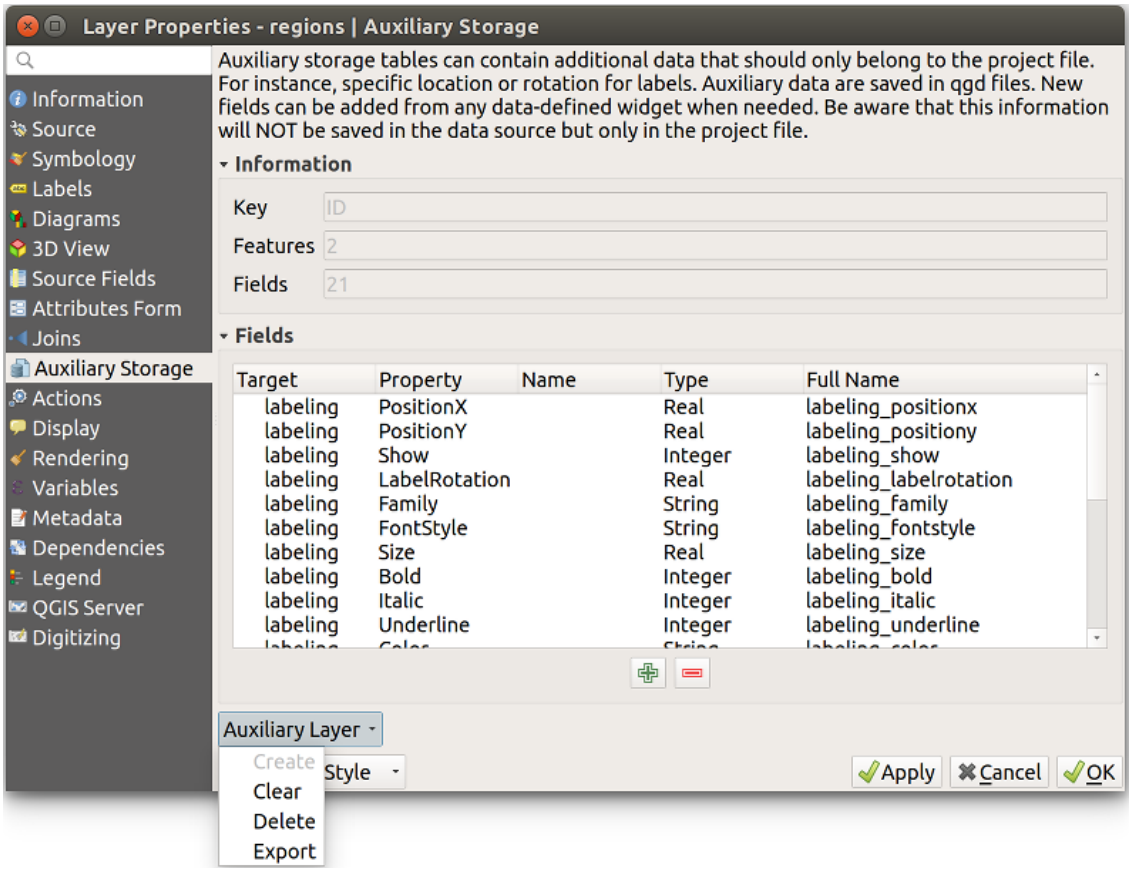
Auxiliary fields in the attribute table will appear like the following image:



12.56 Fig. : Form with auxiliary fields

Management

The *Auxiliary Layer* menu allows you to manage the auxiliary fields:



12.57 Fig. : Auxiliary layer management

The first item *Create* is disabled in this case because the auxiliary layer is already created. But in case of a fresh work, you can use this action to create an auxiliary layer. As explained in *Labeling*, a primary key will be needed then.

The *Clear* action allows to keep all auxiliary fields, but remove their contents. This way, the number of features using these fields will fall to `0`.

The *Delete* action completely removes the auxiliary layer. In other words, the corresponding table is deleted from the underlying SQLite database and properties customization are lost.

Finally, the *Export* action allows to save the auxiliary layer as a *new vector layer*. Note that geometries are not stored in auxiliary storage. However, in this case, geometries are exported from the original data source too.

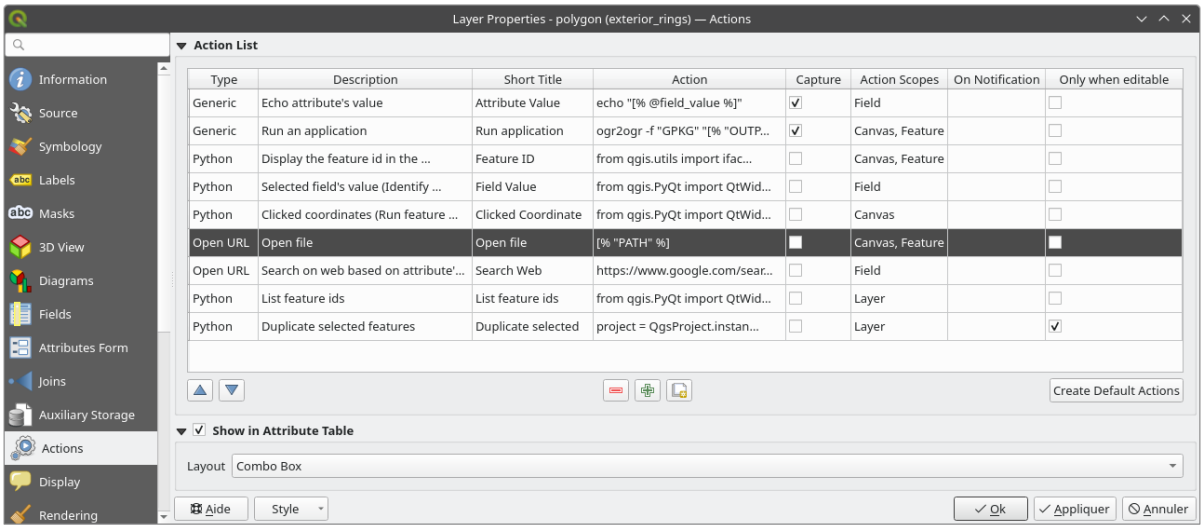
Auxiliary storage database

When you save your project with the `.qgs` format, the SQLite database used for auxiliary storage is saved at the same place but with the extension `.qgd`.

For convenience, an archive may be used instead thanks to the `.qgz` format. In this case, `.qgd` and `.qgs` files are both embedded in the archive.

12.1.12 Actions Properties

The *Actions* tab provides the ability to perform an action based on the attributes of a feature. This can be used to perform any number of actions, for example, running a program with arguments built from the attributes of a feature or passing parameters to a web reporting tool.



12.58 Fig. : Overview action dialog with some sample actions

Actions are useful when you frequently want to run an external application or view a web page based on one or more values in your vector layer. They are of different types and can be used like this:

- *Generic*, *macOS*, *Windows* and *Unix* actions start an external process.
- *Python* actions execute a Python expression.
- *Generic* and *Python* actions are visible everywhere.
- *macOS*, *Windows* and *Unix* actions are visible only on the respective platform (i.e., you can define three „Edit“ actions to open an editor and the users can only see and execute the one „Edit“ action for their platform to run the editor).
- *Open URL*: Uses a HTTP GET request to open a provided URL.
- *Submit URL (urlencoded or JSON)*: Same as the *Open URL* action but using a HTTP POST request. Data are posted to a URL, using „application/x-www-form-urlencoded“ or „application/json“ if the body is a valid JSON.

  An example of action call could be:

  ```
  http://localhost:8000?/[% url_encode(map('file', 'index.php')) %]
  ```

- *Submit URL (multipart)*: Same as the *Open URL* action but using a HTTP POST request. Data are posted to a URL, using „multipart/form-data“.

There are several examples included in the dialog. You can load them by clicking on *Create Default Actions*. To edit any of the examples, double-click its row. One example is performing a search based on an attribute value. This concept is used in the following discussion.

The *Show in Attribute Table* allows you to display in the attribute table dialog the checked feature-scoped actions, either as *Combo Box* or as *Separate Buttons* (see *Configuring the columns*).

Defining Actions

To define an attribute action, open the vector *Layer Properties* dialog and click on the *Actions* tab. In the *Actions* tab, click the Add a new action to open the *Edit Action* dialog.

Select the action *Type* and provide a descriptive name for the action. The action itself must contain the name of the application that will be executed when the action is invoked. You can add one or more attribute field values as arguments to the application. When the action is invoked, any set of characters that start with a `%` followed by the name of a field will be replaced by the value of that field. The special characters `%%` will be replaced by the value of the field that was selected from the identify results or attribute table (see *Using Actions*). Double quote marks can be used to group text into a single argument to the program, script or command. Double quotes will be ignored if preceded by a backslash.

Actions can invoke a single process, with arguments, so Boolean operators (such as `&`, `&&`, `;`, `|`) will not work. In UNIX-like operating systems multiple commands can be executed via `bash -c`.

The *Action Scopes* allows you to define where the action should be available. You have following choices:

1. *Field*: action is available when right click in the cell within the attribute table, in the feature form and in the default action button of the main toolbar.
2. *Feature*: action is available when right click in the cell within the attribute table.
3. *Canvas*: action is available in the main action button in the toolbar.
4. *Form*: action is available only in a feature form designed using the *drag-and-drop* mode.
5. *Layer*: action is available in the action button in the attribute table toolbar. Be aware that this type of action involves the entire layer and not the single features.

If you have field names that are substrings of other field names (e.g., `col1` and `col10`), you should indicate that by surrounding the field name (and the % character) with square brackets (e.g., `[%col10]`). This will prevent the `%col10` field name from being mistaken for the `%col1` field name with a `0` on the end. The brackets will be removed by QGIS when it substitutes in the value of the field. If you want the substituted field to be surrounded by square brackets, use a second set like this: `[[%col10]]`.

Using the *Identify Features* tool, you can open the *Identify Results* dialog. It includes a *(Derived)* item that contains information relevant to the layer type. The values in this item can be accessed in a similar way to the other fields by proceeding the derived field name with `(Derived).`. For example, a point layer has an `X` and `Y` field, and the values of these fields can be used in the action with `%(Derived).X` and `%(Derived).Y`. The derived attributes are only available from the *Identify Results* dialog box, not the *Attribute Table* dialog box.

Two example actions are shown below:

- `konqueror https://www.google.com/search?q=%nam`
- `konqueror https://www.google.com/search?q=%%`

In the first example, the web browser konqueror is invoked and passed a URL to open. The URL performs a Google search on the value of the `nam` field from our vector layer. Note that the application or script called by the action must be in the path, or you must provide the full path. To be certain, we could rewrite the first example as: `/opt/kde3/bin/konqueror https://www.google.com/search?q=%nam`. This will ensure that the konqueror application will be executed when the action is invoked.

The second example uses the %% notation, which does not rely on a particular field for its value. When the action is invoked, the %% will be replaced by the value of the selected field in the identify results or attribute table.

Duplicating Actions

QGIS allows you to duplicate existing actions. To duplicate an attribute action, open the vector *Layer Properties* dialog and click on the *Actions* tab. In the *Actions* tab, click the Duplicate an action to open the *Duplicate Action* dialog. You must have selected at least one existing action in order to create a duplicate.

In the dialogue that appears, make any changes that are necessary. See *Defining Actions* for further information. Once finished, press *OK* to create a duplicate of the action with any changes that you made. If you did not edit the description, or if you changed it to be identical to the description of any other existing action, „_1“ will be added to the end of it.

Using Actions

QGIS offers many ways to execute actions you enabled on a layer. Depending on their settings, they can be available:

- in the drop-down menu of Run Feature Action button from the *Attributes toolbar* or *Attribute table* dialog;
- when right-clicking a feature with the Identify Features tool (see *Geoobjektų identifikavimas* for more information);
- from the *Identify Results* panel, under the *Actions* section;
- as items of an *Actions* column in the *Attribute Table* dialog.

If you are invoking an action that uses the `%%` notation, right-click on the field value in the *Identify Results* dialog or the *Attribute Table* dialog that you wish to pass to the application or script.

Here is another example that pulls data out of a vector layer and inserts it into a file using bash and the `echo` command (so it will only work on or perhaps). The layer in question has fields for a species name `taxon_name`, latitude `lat` and longitude `long`. We would like to be able to make a spatial selection of localities and export these field values to a text file for the selected record (shown in yellow in the QGIS map area). Here is the action to achieve this:

```
bash -c "echo \"%taxon_name %lat %long\" >> /tmp/species_localities.txt"
```

After selecting a few localities and running the action on each one, opening the output file will show something like this:

```
Acacia mearnsii -34.0800000000 150.0800000000
Acacia mearnsii -34.9000000000 150.1200000000
Acacia mearnsii -35.2200000000 149.9300000000
Acacia mearnsii -32.2700000000 150.4100000000
```

As an exercise, we can create an action that does a Google search on the `lakes` layer. First, we need to determine the URL required to perform a search on a keyword. This is easily done by just going to Google and doing a simple search, then grabbing the URL from the address bar in your browser. From this little effort, we see that the format is https://www.google.com/search?q=QGIS, where `QGIS` is the search term. Armed with this information, we can proceed:

1. Make sure the `lakes` layer is loaded.
2. Open the *Layer Properties* dialog by double-clicking on the layer in the legend, or right-click and choose *Properties* from the pop-up menu.
3. Click on the *Actions* tab.
4. Click Add a new action.
5. Choose the *Open URL* action type,
6. Enter a name for the action, for example `Google Search`.
7. Additionally you can add a *Short Name* or even an *Icon*.

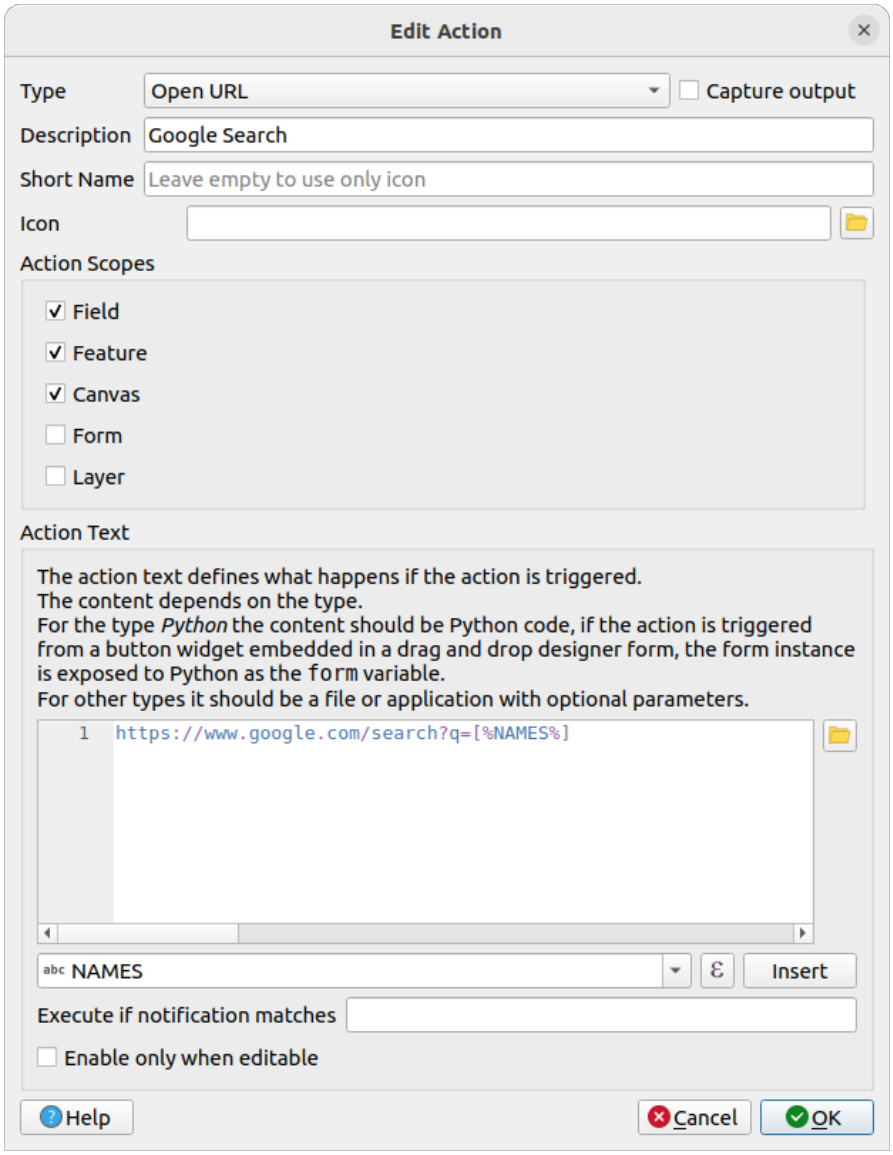
8. Choose the *Action Scopes*. See *Defining Actions* for further information. Leave the default settings for this example.
9. For the action, add the URL used for doing a Google search, up to but not including the search term: `https://www.google.com/search?q=`
10. The text in the *Action* field should now look like this:

```
https://www.google.com/search?q=
```

11. Click on the drop-down box containing the field names for the `lakes` layer. It's located just to the left of the *Insert* button.
12. From the drop-down box, select *NAMES* and click *Insert*.
13. Your action text now looks like this:

```
https://www.google.com/search?q=[%NAMES%]
```

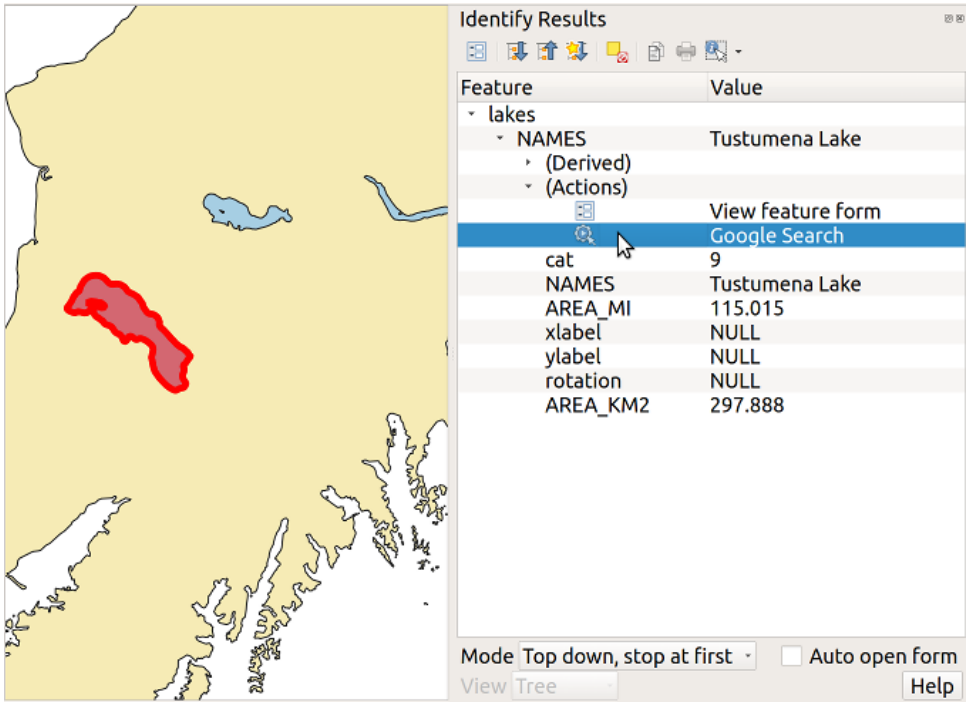
14. To finalize and add the action, click the *OK* button.



12.59 Fig. : Edit action dialog configured with the example

This completes the action, and it is ready to use.

Close the *Layer Properties* dialog and zoom in to an area of interest. Make sure the `lakes` layer is active and identify a lake. In the result box you'll now see that our action is visible:



12.60 Fig. : Select feature and choose action

When we click on the action, it brings up our default browser and navigates to the URL https://www.google.com/search?q=Tustumena. It is also possible to add further attribute fields to the action. Therefore, you can add a + to the end of the action text, select another field and click on *Insert Field*. In this example, there is just no other field available that would make sense to search for.

You can define multiple actions for a layer, and each will show up in the *Identify Results* dialog.

You can also invoke actions from the attribute table by selecting a row and right-clicking, then choosing the action from the pop-up menu.

There are all kinds of uses for actions. For example, if you have a point layer containing locations of images or photos along with a file name, you could create an action to launch a viewer to display the image. You could also use actions to launch web-based reports for an attribute field or combination of fields, specifying them in the same way we did in our Google search example.

We can also make more complex examples, for instance, using **Python** actions.

Usually, when we create an action to open a file with an external application, we can use absolute paths, or eventually relative paths. In the second case, the path is relative to the location of the external program executable file. But what about if we need to use relative paths, relative to the selected layer (a file-based one, like Shapefile or SpatiaLite)? The following code will do the trick:

```
command = "firefox"
imagerelpath = "images_test/test_image.jpg"
layer = qgis.utils.iface.activeLayer()
import os.path
layerpath = layer.source() if layer.providerType() == 'ogr'
  else (qgis.core.QgsDataSourceURI(layer.source()).database()
    if layer.providerType() == 'spatialite' else None)
path = os.path.dirname(str(layerpath))
image = os.path.join(path,imagerelpath)
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
import subprocess
subprocess.Popen( [command, image ] )
```

We just have to remember that the action is one of type *Python* and the *command* and *imagerelpath* variables must be changed to fit our needs.

But what about if the relative path needs to be relative to the (saved) project file? The code of the Python action would be:

```
command = "firefox"
imagerelpath = "images_test/test_image.jpg"
projectpath = qgis.core.QgsProject.instance().fileName()
import os.path
path = os.path.dirname(str(projectpath)) if projectpath != '' else None
image = os.path.join(path, imagerelpath)
import subprocess
subprocess.Popen( [command, image ] )
```

Another Python action example is the one that allows us to add new layers to the project. For instance, the following examples will add to the project respectively a vector and a raster. The names of the files to be added to the project and the names to be given to the layers are data driven (*filename* and *layername* are column names of the table of attributes of the vector where the action was created):

```
qgis.utils.iface.addVectorLayer('/yourpath/[% "filename" %].shp',
  '[% "layername" %]', 'ogr')
```

To add a raster (a TIF image in this example), it becomes:

```
qgis.utils.iface.addRasterLayer('/yourpath/[% "filename" %].tif',
  '[% "layername" %]')
```

12.1.13 Display Properties

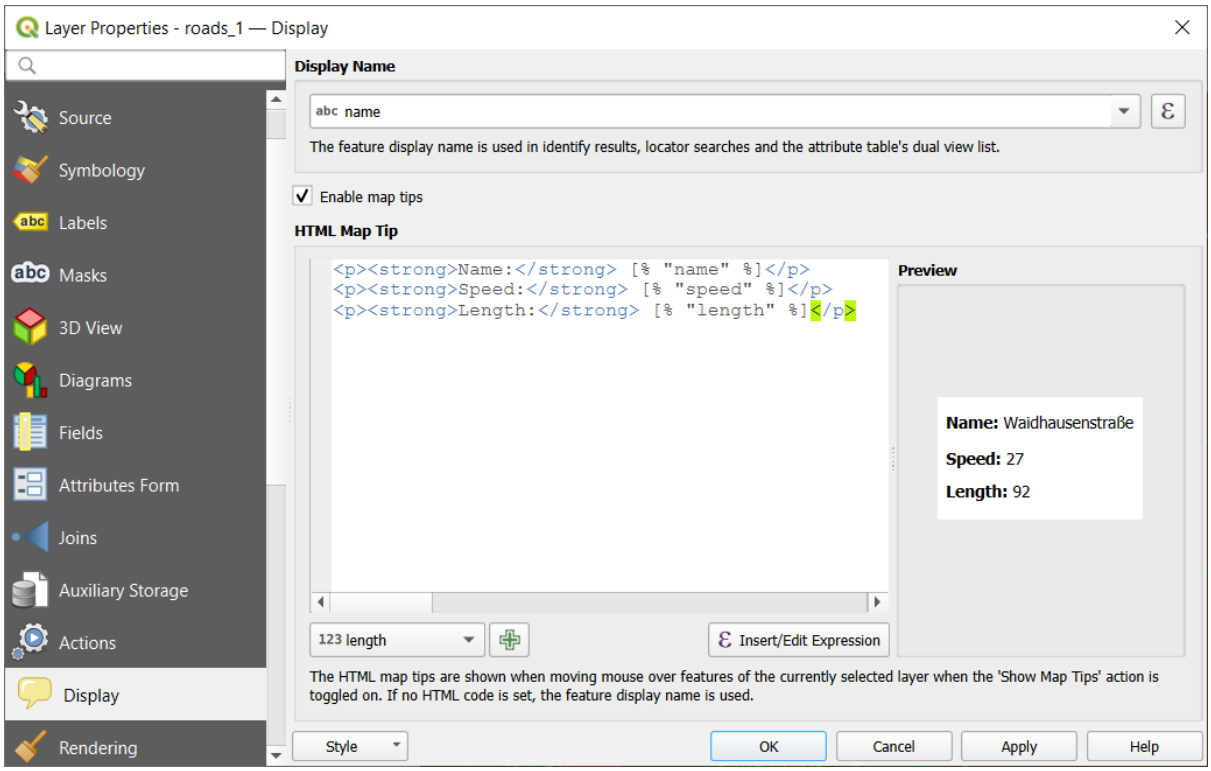
The *Display* tab helps you configure fields to use for feature identification:

- The *Display name*: based on a field or an *expression*. By default, this is set to the first field in the layer if no field with `<name>` component exists. This is used as:
  - the label shown on top of the feature information in the *Identify tool* results
  - the field used in the *locator bar* when looking for features in all layers
  - the feature identifier in the attribute table *form view*
  - the feature identifier when the map or layout is exported to a layered output format such as Geospatial PDF
  - the map tip information, i.e. the message displayed in the map canvas when hovering over a feature of the active layer with the Show Map Tips icon pressed. Applicable when *Enable Map Tips* is active and no *HTML Map Tip* is set.
- *Enable Map Tips* controls whether to display map tips for the layer
- The *HTML Map Tip* provides a complex and full HTML text editor for map tips, mixing QGIS expressions and html styles and tags (multiline, fonts, images, hyperlink, tables, …). You can check the result of your code sample in the *Preview* frame (also convenient for previewing the *Display name* output). Additionally, you can select and edit existing expressions using the *Insert/Edit Expression* button.

Pastaba

Understanding the *Insert/Edit Expression* button behavior

If you select some text within an expression (between „[%“ and „%]“), or if no text is selected but the cursor is inside an expression, the whole expression will be automatically selected for editing. If the cursor or a selected text is outside an expression, the dialog opens with the selection.



12.61 Fig. : HTML code for map tip

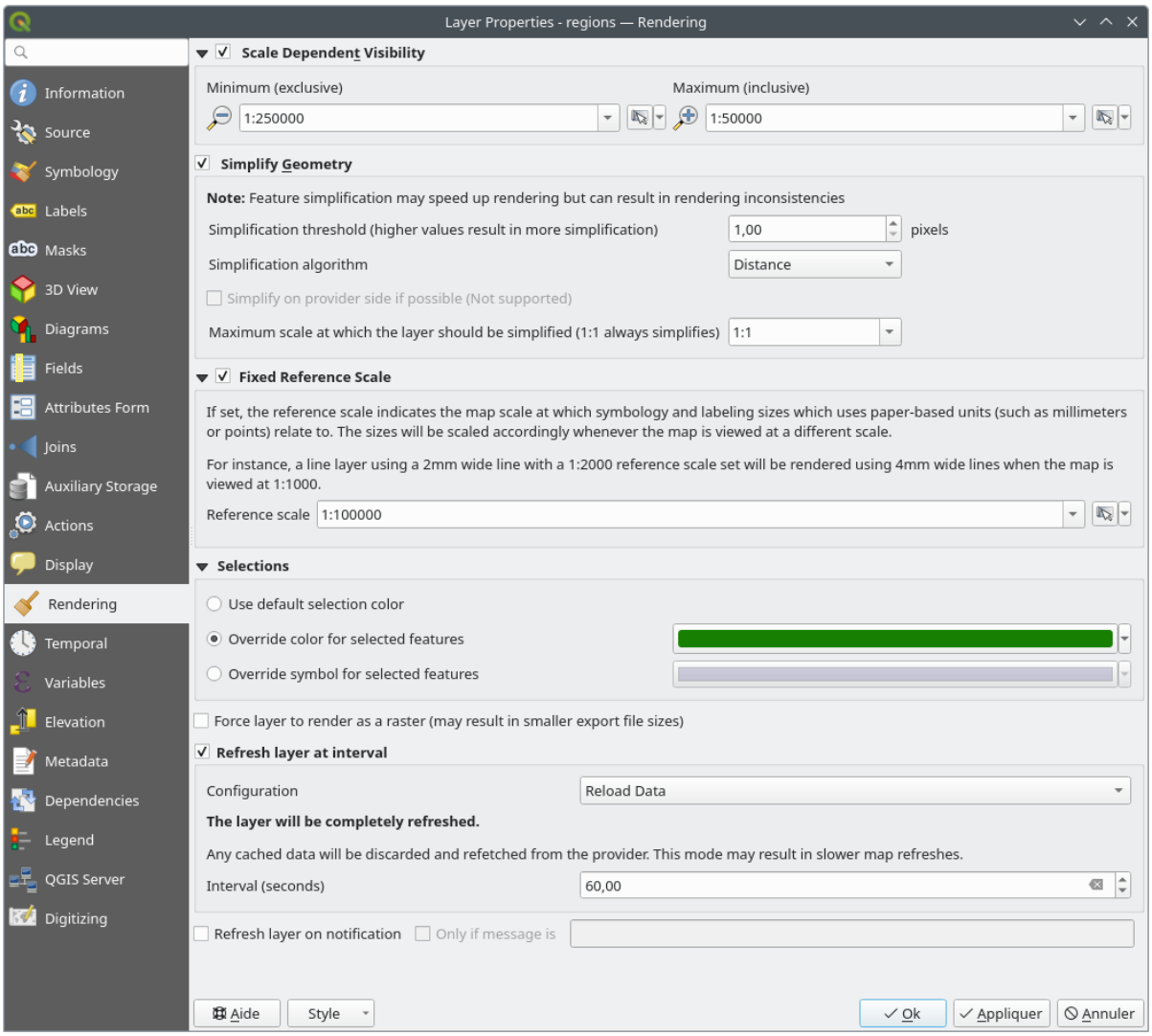
To display map tips:

1. Select the menu option *View* ► *Show Map Tips* or click on the Show Map Tips icon of the *Attributes Toolbar*.
2. Make sure that the layer you target is active and has the *Enable Map Tips* property checked.
3. Move over a feature, and the corresponding information will be displayed over.

Map tip is a cross-layer feature meaning that once activated, it stays on and applies to any map tip enabled layer in the project until it’s toggled off.

12.62 Fig. : Map tip made with HTML code

12.1.14 Rendering Properties



12.63 Fig. : Layer Rendering Properties dialog

The *Rendering* tab offers following properties:

- Under *Scale dependent visibility*, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scales, defining a range of scales in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See *Nuo mastelio priklausomo matomumo parinkimas* for more information.

  > **Pastaba**
  >
  > You can also activate scale dependent visibility on a layer from within the *Layers* panel: right-click on the layer and in the contextual menu, select *Set Layer Scale Visibility*.

- QGIS offers support for on-the-fly feature generalisation. This can improve rendering times when drawing many complex features at small scales. This feature can be enabled or disabled in the layer settings using the *Simplify geometry* option. There is also a global setting that enables generalisation by default for newly added layers (see *global simplification* for more information).

> **Pastaba**
>
> Feature generalisation may introduce artefacts into your rendered output in some cases. These may include slivers between polygons and inaccurate rendering when using offset-based symbol layers.

- The *Fixed reference scale* indicates the map scale at which symbology and labeling sizes which uses paper-based units (such as millimeters or points) relate to. The sizes will be scaled accordingly whenever the map is viewed at a different scale.

  For instance, a line layer using a 2mm wide line with a 1:2000 *Reference scale* set will be rendered using 4mm wide lines when the map is viewed at 1:1000.

- The *Selections* group allows you to control whether a specific color or symbol should be used in place of the defaults (*Project properties* ► *General* ► *Selection color*) for a particular layer. This is useful to improve the visibility of selected features with certain symbology:

  - *Use default selection color*
  - *Override color for selected features*: e.g. if the layer uses a yellow color by default and the standard yellow selection is not visible.
  - *Override symbol for selected features*: e.g. if a line layer uses a thin symbol, and coloring the lines might not make them visible enough, overriding the symbol with a thicker line can help. Also, if the layer uses raster symbology or gradient fills/lines/shapeburst with color ramp symbology, then the default selection color is not applied at all; being able to set a specific simpler symbol to use for selected features in the layer can help.

- Rendering extremely detailed layers (e.g. polygon layers with a huge number of nodes), can cause layout exports in PDF/SVG format to be huge as all nodes are included in the exported file. This can also make the resultant file very slow to work with/open in other programs.

  Checking *Force layer to render as raster* forces these layers to be rasterised so that the exported files won't have to include all the nodes contained in these layers and the rendering is therefore sped up.

  You can also do this by forcing the layout to export as a raster, but that is an all-or-nothing solution, given that the rasterisation is applied to all layers. Alternatively, you can rely on geometry simplification in *layout export settings*.

- *Refresh layer at interval*: controls whether and how regular a layer can be refreshed. Available *Configuration* options are:

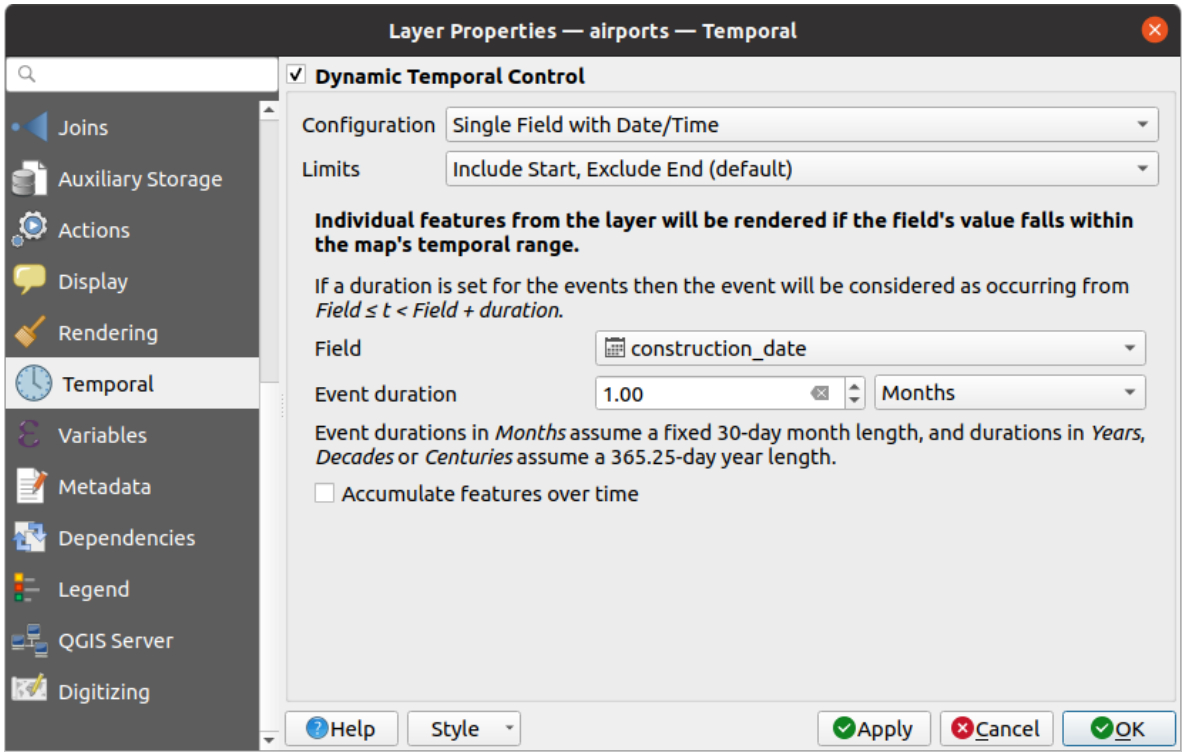
  - *Reload data*: the layer will be completely refreshed. Any cached data will be discarded and refetched from the provider. This mode may result in slower map refreshes.
  - *Redraw layer only*: this mode is useful for animation or when the layer's style will be updated at regular intervals. Canvas updates are deferred in order to avoid refreshing multiple times if more than one layer has an auto update interval set.

  It is also possible to set the *Interval (seconds)* between consecutive refreshments.

- Depending on the data provider (e.g. PostgreSQL), notifications can be sent to QGIS when changes are applied to the data source, out of QGIS. Use the *Refresh layer on notification* option to trigger an update. You can also limit the layer refresh to a specific message set in the *Only if message is* text box.

12.1.15 Laiko savybės

The *Temporal* tab provides options to control the rendering of the layer over time. Such dynamic rendering requires the *temporal navigation* to be enabled over the map canvas.



12.64 Fig. : Vector layer temporal properties dialog

Check the *Dynamic Temporal Control* option to configure the vector layer temporal rendering. Depending on the structure of your dataset, you may want to use one of the provided *Configuration* options:

- *Fixed time range*: all the features are rendered if the map canvas temporal frame overlaps the given *Start date* and *End date* range.
- *Single field with date/time*: features are rendered if their *Field*'s value falls within the map canvas temporal frame. An *Event duration* can be set. With checking the *Accumulate features over time* option, all features which occur before or within the map's temporal range will continue to be rendered. The event duration is thus ignored.
- *Separate fields for start and end date/time*: features are rendered if the range specified by their *Start field* and *End field* values overlaps the map canvas temporal.
- *Separate fields for start and event duration*: features are rendered if the range defined by their *Start field* and *Event duration field* values overlaps the map canvas temporal.
- *Start and end date/time from expressions*: features are rendered if the time range specified by the fields *Start expression* and *End expression* overlaps the map canvas temporal.
- *Redraw layer only*: the layer is redrawn at each new animation frame but no time-based filtering is applied to the features. It's useful when the layer uses time-based expression values for renderer settings (e.g. data-defined symbology).

It is also possible to set the *Limits* of the features time range as:

- *Include start, exclude end*
- *Include start, include end*

12.1.16 Kintamųjų savybės

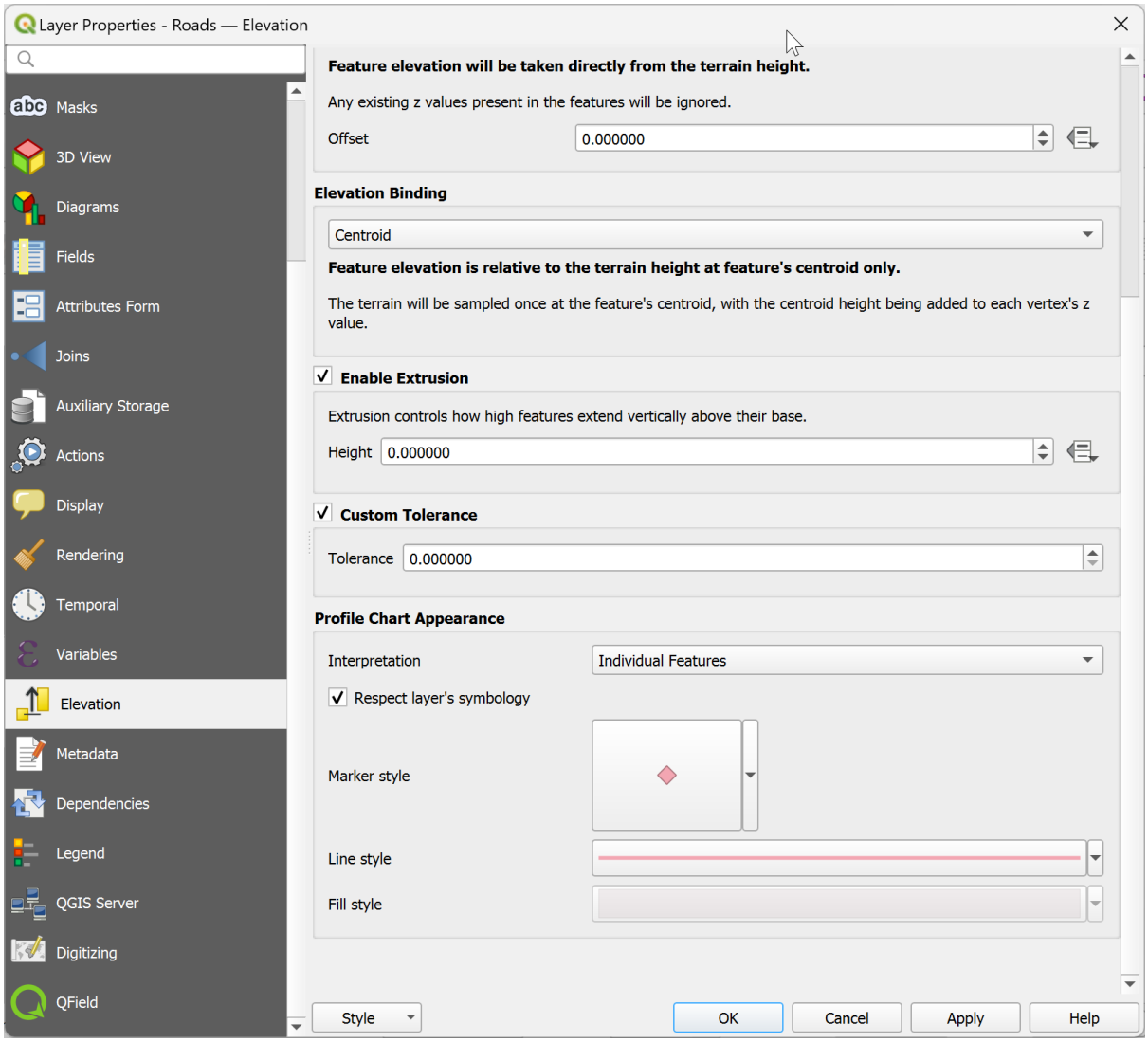
The *Variables* tab lists all the variables available at the layer's level (which includes all global and project's variables).

It also allows the user to manage layer-level variables. Click the button to add a new custom layer-level variable. Likewise, select a custom layer-level variable from the list and click the button to remove it.

More information on variables usage in the General Tools *Reikšmių saugojimas kintamuosiuose* section.

12.1.17 Aukščio savybės

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view* and its appearance in the *profile tool charts*. Specifically, you can set:



12.65 Fig. : Vector layer elevation properties dialog

- *Vertical Reference System*: If the CRS of your vector layer is a compound one (including a Z dimension), then the vertical CRS used for the layer will be the vertical component of the layer CRS. In this case, you cannot

manually set a different vertical CRS. If your layer CRS is horizontal (2D), then you can select a specific vertical CRS by clicking on the Select CRS. Vertical reference systems are supported for vector layers by:

  - *Elevation profiles*
  - *Identify Tool Results*
  - *3D map views*

- *Elevation Clamping*: defines how and whether the features altitude should be:
  - *Clamped to terrain*: takes elevation directly from the terrain height and ignores any existing Z values in the features. A data-defined *Offset* value from the terrain can also be filled.
  - *Relative to terrain*: any existing Z values in the features are added to the terrain height. A *Scale* factor followed by a data-defined *Offset* can be used to adjust the elevation. This option is not available for 2D geometry layers.
  - *Absolute*: ignores the terrain height and directly takes Z values from the features for the elevation. A *Scale* factor followed by a data-defined *Offset* can be used to adjust the elevation. For 2D geometry layers (with no Z values), a data-defined *Base height* can instead be set.
- *Enable extrusion*: you can set a *Height* to control how high features vertically extend above their base. This is convenient to indicate that a 2D geometry layers, e.g. a polygon building footprints layer, actually represents 3D objects.
- *Custom tolerance*: allows you to override the global sampling tolerance defined in the elevation profile settings, specifically for this vector layer. When enabled, you can set *Tolerance* that determines how the layer is sampled along the profile line. This is useful when you need finer or coarser sampling than the global default, for example, to improve performance or accuracy for complex geometries.
- *Elevation Binding*: only relevant when combining an *Elevation clamping* relying on the terrain with a line or polygon layer, this option controls how feature elevation is set relative to the terrain height. The terrain can be sampled:
  - at the feature's *Centroid*, with the centroid height being added to each vertex's z value
  - at every individual *Vertex* before being added to the vertex's z value
- *Profile Chart Appearance*: controls how features are rendered when drawing a profile chart. Two main *Interpretation* modes are available:
  - as *Individual features*: samples discrete positions where the cross section profile line intersects the vector features. That intersection can be represented as point, line or surface depending on the layer type and whether an extrusion is applied.

    With checking *Respect layer symbology*, features will be rendered on the profile chart with their corresponding *layer styling* (allowing e.g. categorized classes to be visible on the profile chart). If the profile symbol type does not match the layer's renderer symbol types, only the symbol color from the renderer is applied to the profile symbol.

    Depending on the layer settings, profile symbols can be represented with a custom style, using:

    * *Marker style*: for non-extruded point and line features, and for non-extruded polygon features touched by the profile line
    * *Line style*: for extruded point and line features, and for non-extruded polygon features intersected by the profile line
    * *Fill style*: for extruded polygon features
  - as *Continuous Surface (e.g. contours)*: the elevation chart will be rendered as a surface instead of separate features by joining the sampled elevation results into a continuous line. This can enhance visualisation and is designed for vector layers which represent a continuous elevation surface, e.g. contour lines or surveyed elevation points. The profile *Style* can be set as:

    * a *Line* with a specific *Line style*

    * an elevation surface rendered using a fill symbol either above (*Fill above*) or below (*Fill below*) the elevation curve line. The surface symbology is represented using:
        - a *Fill style*
        - and a *Limit*: the maximum (respectively minimum) altitude determining how high the fill surface will be

    Moreover, you can check *Show markers at sampled points* to make them visible over the interpretation line and assign them a *Marker style*.

12.1.18 Metaduomenų savybės

The *Metadata* tab provides you with options to create and edit a metadata report on your layer. See *Metaduomenys* for more information.

12.1.19 Dependencies Properties

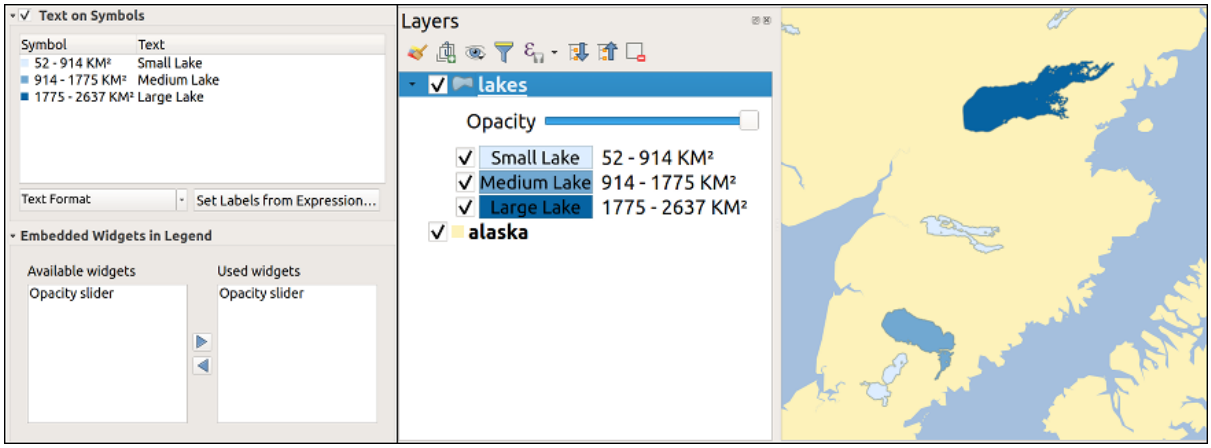
The *Dependencies* tab allows to declare data dependencies between layers. A data dependency occurs when a data modification in a layer, not by direct user manipulation, may modify data of other layers. This is the case for instance when geometry of a layer is updated by a database trigger or custom PyQGIS scripting after modification of another layer's geometry.

In the *Dependencies* tab, you can select any layers which may externally alter the data in the current layer. Correctly specifying dependent layers allows QGIS to invalidate caches for this layer when the dependent layers are altered.

12.1.20 Legend Properties

The *Legend* properties tab provides you with advanced settings for the *Layers panel* and/or the *print layout legend*. These options include:

- Depending on the symbology applied to the layer, you may end up with several entries in the legend, not necessarily readable/useful to display. The *Legend placeholder image* helps you *select an image* for replacement, displayed both in the *Layers* panel and the print layout legend.
- *Show label legend*: Displays overviews of the different label settings as entries in the legends. The *label style* is previewed along with the description.
- *Text on symbols*: In some cases it can be useful to add extra information to the symbols in the legend. With this frame, you can affect to any of the symbols used in the layer symbology a text that is displayed over the symbol, in both *Layers* panel and print layout legend. This mapping is done by typing each text next to the symbol in the table widget or filling the table using the *Set Labels from Expression* button. Text appearance is handled through the font and color selector widgets of the *Text Format* button.



12.66 Fig. : Setting text on symbols (left) and its rendering in the *Layers* panel (right)

- a list of widgets you can embed within the layer tree in the Layers panel. The idea is to have a way to quickly access some actions that are often used with the layer (setup transparency, filtering, selection, style or other stuff…).

  By default, QGIS provides transparency widget but this can be extended by plugins registering their own widgets and assign custom actions to layers they manage.

12.1.21 QGIS serverio savybės

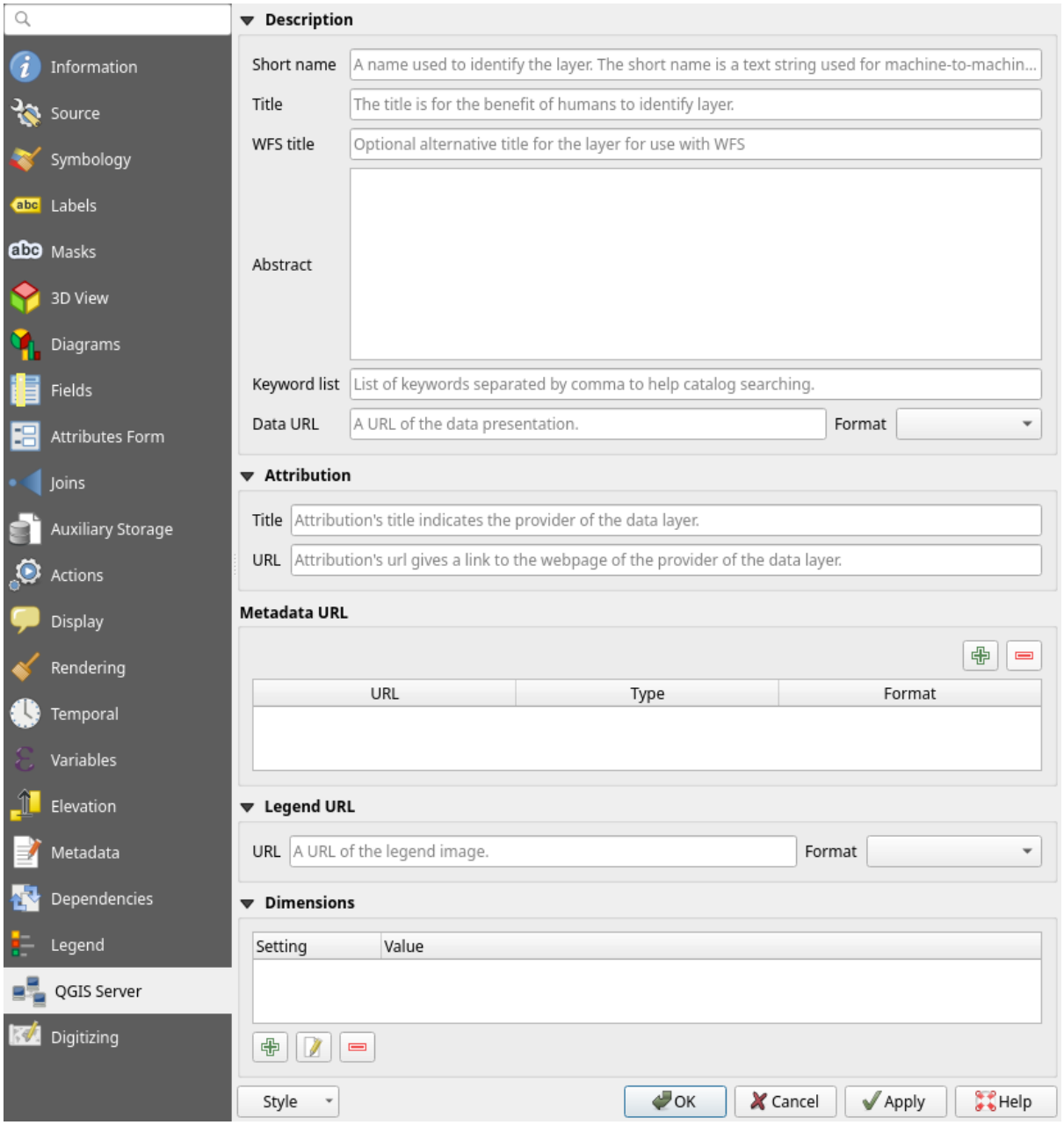
The *QGIS Server* tab consists of *Description*, *Attribution*, *Metadata URL*, and *Legend URL* sections.

From the *Description* section, you can change the *Short name* used to reference the layer in requests (to learn more about short names, read services_basics_short_name). You can also add or edit a *Title*, an alternative *WFS Title* and *Abstract* for the layer, or define a *Keyword list* here. These keyword lists can be used in a metadata catalog. If you want to use a title from an XML metadata file, you have to fill in a link in the *Data URL* field.

Use *Attribution* to get attribute data from an XML metadata catalog.

In *Metadata URL*, you can add the general paths to the XML metadata catalog. This information will be saved in the QGIS project file for subsequent sessions and will be used for QGIS Server.

In the *Legend URL* section, you can provide the url of a legend image in the url field. You can use the Format drop-down option to apply the appropriate format of the image. Currently png, jpg and jpeg image formats are supported.

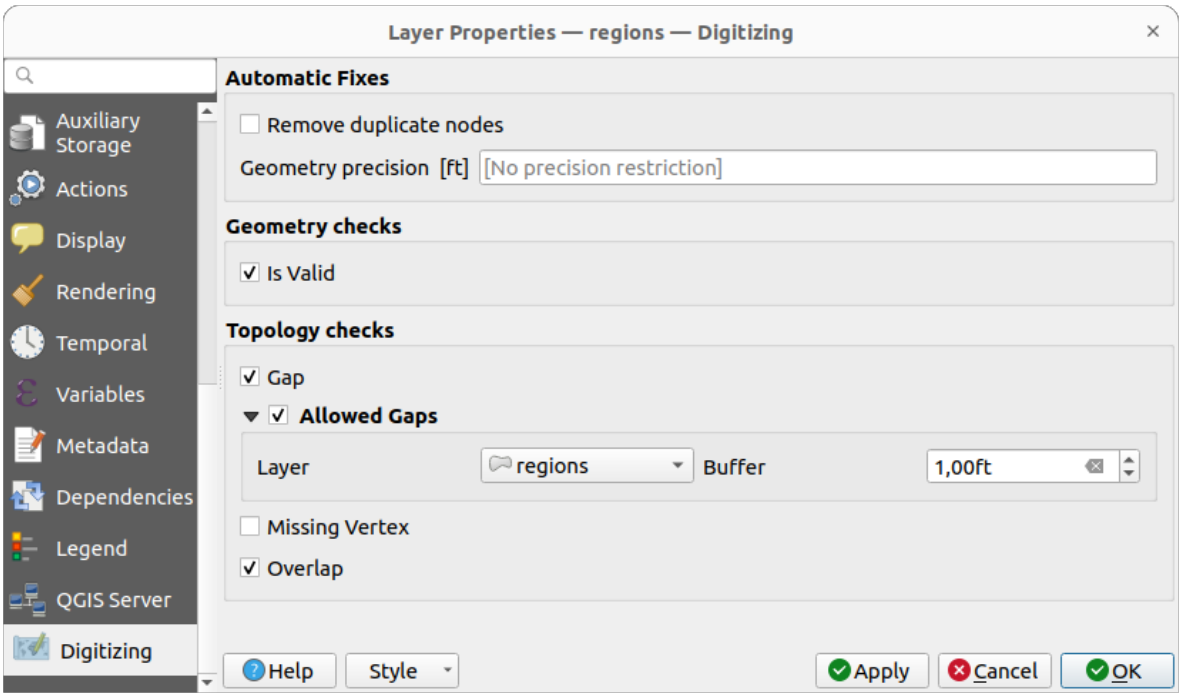


12.67 Fig. : QGIS Server tab in vector layers properties dialog

To learn more about QGIS Server, read the QGIS-Server-manual.

12.1.22 Digitizing Properties

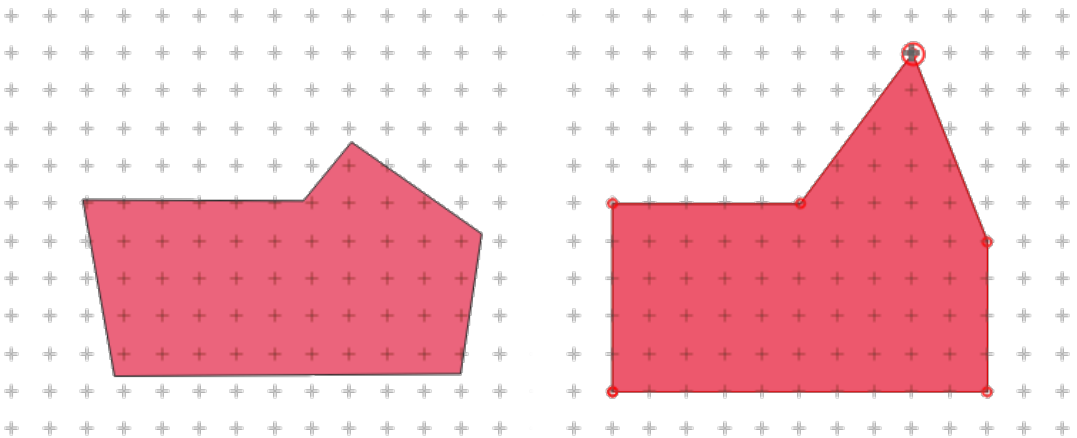
The *Digitizing* tab gives access to options that help to ensure the quality of digitized geometries.



12.68 Fig. : The QGIS Digitizing tab in the vector layers properties dialog

Automatic Fixes

Options in the *Automatic Fixes* section will directly affect the vertices of any geometry which is added or modified. If the *Remove duplicate nodes* option is checked, any two subsequent vertices with exactly the same coordinates will be removed. If the *Geometry precision* is set, all vertices will be rounded to the closest multiple of the configured geometry precision. The rounding will happen in the layer coordinate reference system. Z and M values are not rounded. With many map tools, a grid is shown on the canvas while digitizing.



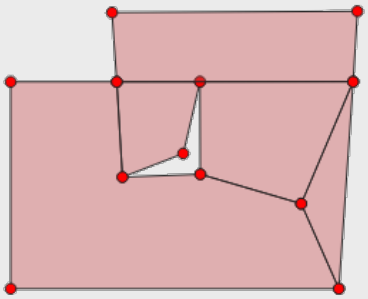
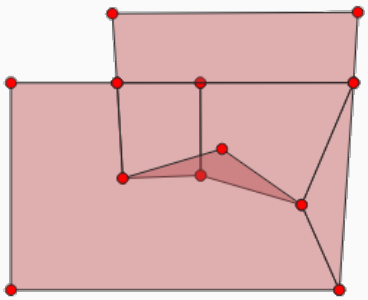
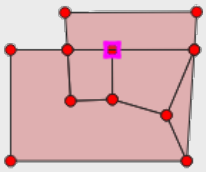
12.69 Fig. : Moving the top vertex snaps all the vertices to the grid

Geometry Checks

In the *Geometry checks* section, additional validations on a per geometry basis can be activated. Immediately after any geometry modification, failures in these checks are reported to the user in the *Geometry validation* panel. As long as a check is failing, it is not possible to save the layer. The *Is valid* check will run basic validity checks like self intersection on geometries.

Topology Checks

In the *Topology checks* section, additional topology validation checks can be activated. Topology checks will be executed when the user saves the layer. Check errors will be reported in the *Geometry validation* panel. As long as validation errors are present, the layer can not be saved. Topology checks are executed in the area of the bounding box of the modified features. Since other features may be present in the same area, topological errors concerning these features are reported as well as errors introduced in the current edit session.

Topology check option	Illustration
The *Gap* check will check for gaps between neighbouring polygons.	
The *Overlap* check will check for overlaps between neighbouring polygons.	
The *Missing vertex* check will check for shared boundaries of neighbouring polygons where one border misses a vertex which is present on the other one.	

Gap check exceptions

Sometimes it is desirable to keep gaps inside an area in a polygon layer that otherwise is fully covered by polygons. For example, a land use layer may have acceptable holes for lakes. It is possible to define areas that are ignored in the gap check. Since gaps inside these areas are allowed, we will refer to them as *Allowed Gaps* areas.

In the options for the gap checks under *Allowed Gaps*, an *Allowed Gaps layer* can be configured.

Whenever the gap check is executed, gaps which are covered by one or more polygons in the *Allowed Gaps Layer* are not reported as topology errors.

It is also possible to configure an additional *Buffer*. This buffer is applied to each polygon on the *Allowed Gaps Layer*. This makes it possible to make the tests less susceptible to small changes in the outlines at the borders of gaps.

When *Allowed Gaps* are enabled, an additional button (*Add Allowed Gap*) for detected gap errors is available in the geometry validation dock, where gaps are reported during digitizing. If the *Add Allowed Gap* button is pushed, a new polygon with the geometry of the detected gap is inserted into the *Allowed Gaps Layer*. This makes it possible to quickly flag gaps as allowed.

Geometry validation panel

The *Geometry Validation* panel is triggered when any of the abovementioned digitizing checks finds an error. The dialog provides you with the list of errors and their description, and you can to browse the list using the keyboard arrows or dedicated arrows.

You'll need to address all the issues before you can save edits to the layer. To do so:

1. Select an error, and it's possible to:
   - *Zoom to Feature(s)*
   - *Zoom to problem*
2. Pick the usual *digitizing tools* to fix the issue.

12.2 Darbas su atributų lentele

Atributų lentelė rodo pažymėto sluoksnio geoobjektų informaciją. Kiekviena lentelės eilutė reprezentuoja geoobjektą (su ar be geometrijos), o kiekviename stulpelyje yra konkreti geoobjekto informacijos dalis. Lentelės geoobjektų galima ieškoti, juos galima pažymėti, perkelti ar net redaguoti.

12.2.1 Įvadas: erdvinės ir neerdvinės lentelės

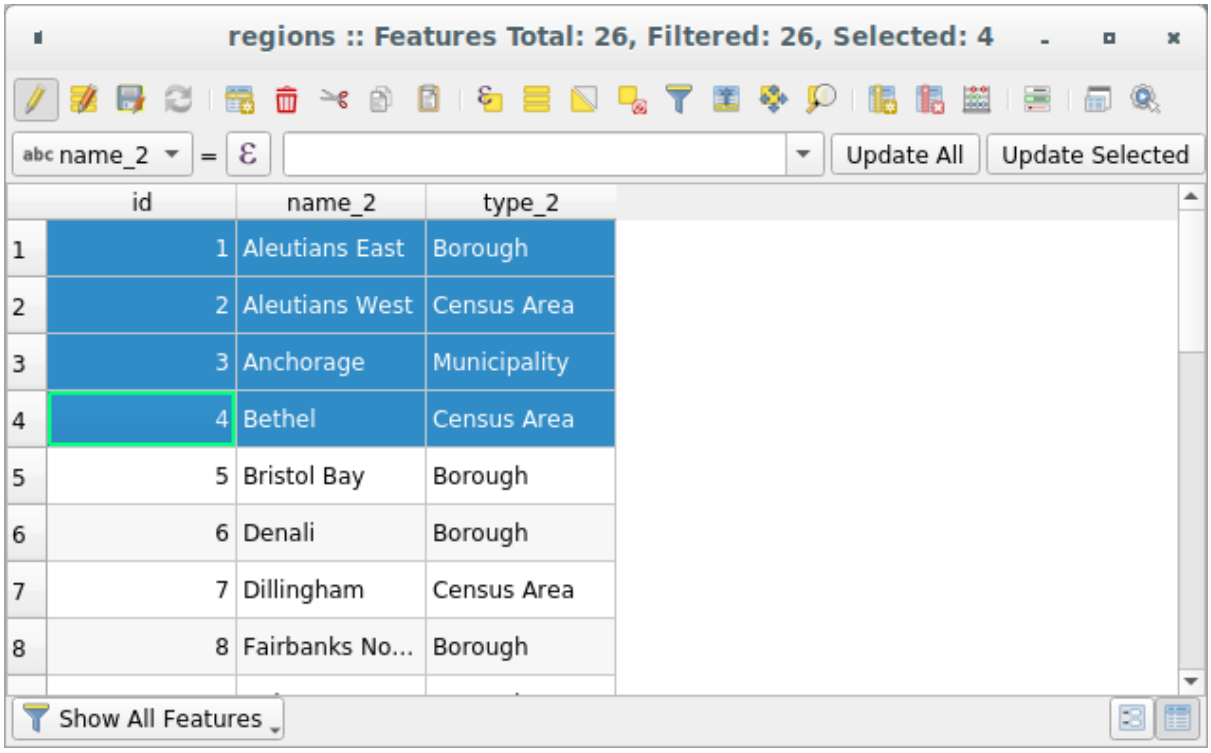
QGIS leidžia įkelti erdvinius ir neerdvinius sluoksnius. Šiuo metu tai reiškia GDAL palaikomas lenteles ir atskirtą tekstą bei PostgreSQL, MS SQL Server, SpatiaLite ir Oracle tiekėjus. Visi įkelti sluoksniai rodomi skydelyje *Sluoksniai*. Priklausomai nuo to, ar sluoksnis yra erdvinis ar ne, jūs galėsite su juo atlikti skirtingus veiksmus.

Non-spatial tables can be browsed and edited using the attribute table view. Furthermore, they can be used for field lookups. For example, you can use columns of a non-spatial table to define attribute values, or a range of values that are allowed, to be added to a specific vector layer during digitizing. Have a closer look at the edit widget in section *Attributes Form Properties* to find out more.

12.2.2 Introducing the attribute table interface

To open the attribute table for a vector layer, activate the layer by clicking on it in the *Sluoksnių skydelis*. Then, from the main *Layer* menu, choose *Open Attribute Table*. It is also possible to right-click on the layer and choose *Open Attribute Table* from the drop-down menu, or to click on the *Open Attribute Table* button in the Attributes toolbar. If you prefer shortcuts, `F6` will open the attribute table. `Shift+F6` will open the attribute table filtered to selected features and `Ctrl+F6` will open the attribute table filtered to visible features.

This will open a new window that displays the feature attributes for the layer (*figure_attributes_table*). According to the setting in *Settings* ► *Options* ► *Data sources* menu, the attribute table will open in a docked window or a regular window. The total number of features in the layer and the number of currently selected/filtered features are shown in the attribute table title, as well as if the layer is spatially limited.



12.70 Fig. : Attribute Table for regions layer

The buttons at the top of the attribute table window provide the following functionality:

12.1 Table : Available Tools

Piktograma	Užrašas	Tikslas	Default Shortcut
	Toggle editing mode	Enable editing functionalities	`Ctrl+E`
	Toggle multi edit mode	Update multiple fields of many features	
	Save Edits	Save current modifications	
	Reload the table		
	Add feature	Add new geometryless feature	
	Delete selected features	Remove selected features from the layer	
	Cut selected features to clipboard		`Ctrl+X`

continues on next page

Table 12.1 – tęsinys iš praeito puslapio

Piktograma	Užrašas	Tikslas	Default Shortcut
	Copy selected features to clipboard		Ctrl+C
	Paste features from clipboard	Insert new features from copied ones	Ctrl+V
	Select features using an Expression		
	Select All	Select all features in the layer	Ctrl+A
	Invert selection	Invert the current selection in the layer	Ctrl+R
	Deselect all	Deselect all features in the current layer	Ctrl+Shift+A
	Filter/Select features using form		Ctrl+F
	Move selected to top	Move selected rows to the top of the table	
	Pan map to the selected rows		Ctrl+P
	Zoom map to the selected rows		Ctrl+J
	New field	Add a new field to the data source	Ctrl+W
	Delete field	Remove a field from the data source	
	Organize columns	Show/hide fields from the attribute table	
	Open field calculator	Update field for many features in a row	Ctrl+I
	Conditional formatting	Enable table formatting	
	Dock attribute table	Allows to dock/undock the attribute table	
	Actions	Lists the actions related to the layer	

Dėmesio

Depending on the format of the data and the GDAL library built with your QGIS version, some tools may not be available.

Below these buttons is the Quick Field Calculation bar (enabled only in *edit mode*), which allows to quickly apply calculations to all or part of the features in the layer. This bar uses the same *expressions* as the Field Calculator (see *Editing attribute values*).

Table view vs Form view

QGIS provides two view modes to easily manipulate data in the attribute table:

- The Table view, displays values of multiple features in a tabular mode, each row representing a feature and each column a field. A right-click on the column header allows you to *configure the table display* while a right-click on a cell provides *interaction with the feature*.

  The attribute table supports `Shift+Mouse Wheel` scrolling in table view mode to switch between vertical and horizontal scrolling movements. This can also be achieved replacing the mouse with the trackpad on macOS.

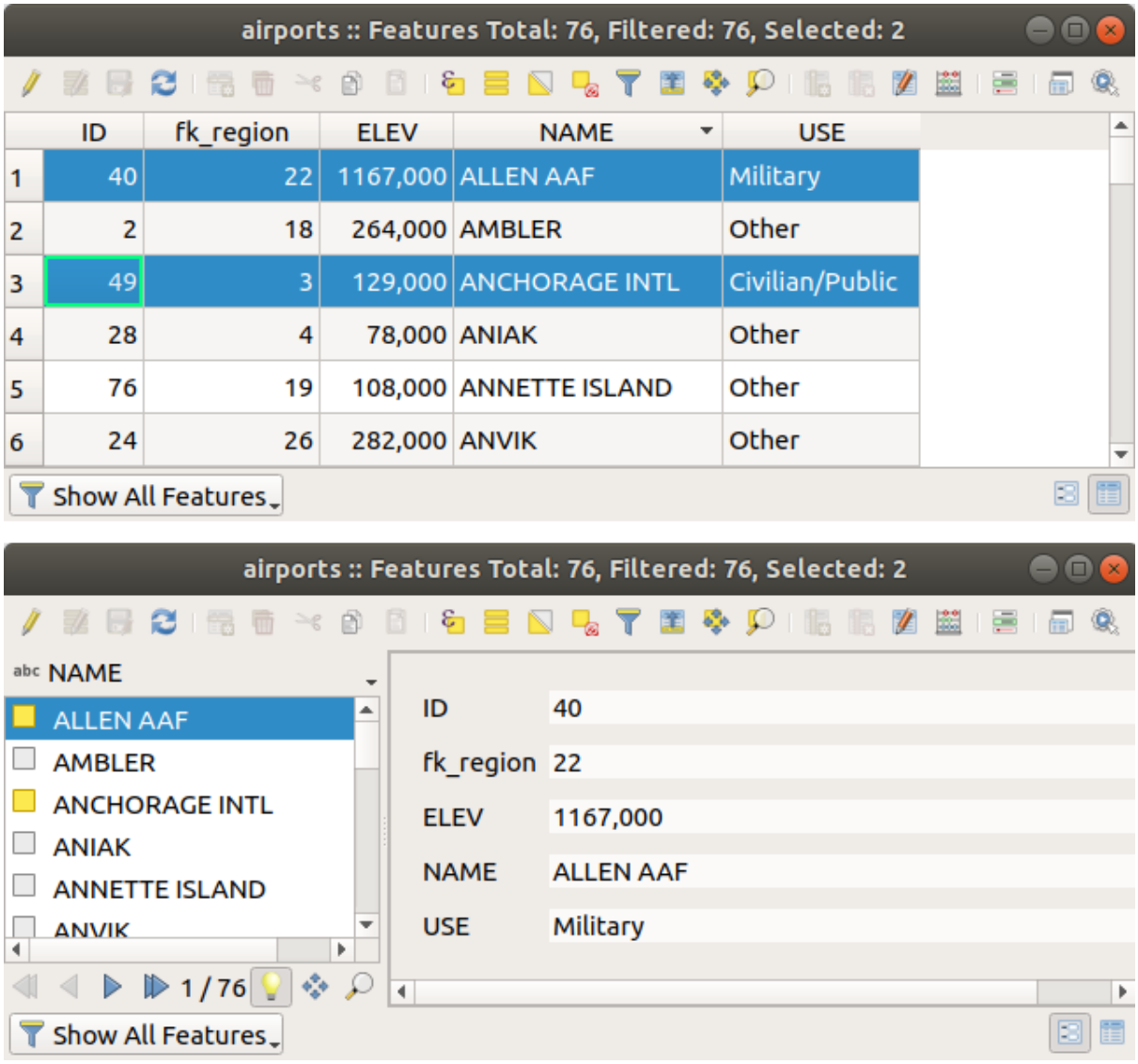
- The Form view shows *feature identifiers* in a first panel and displays only the attributes of the clicked identifier in the second one. There is a pull-down menu at the top of the first panel where the „identifier“ can be specified using an attribute (*Column preview*) or an *Expression*. The pull-down also includes the last 10 expressions for re-use. Form view uses the layer fields configuration (see *Attributes Form Properties*).

You can browse through the feature identifiers with the arrows on the bottom of the first panel. The features attributes update in the second panel as you go. It's also possible to identify or move to the active feature in the map canvas with pushing down any of the button at the bottom:

- Highlight current feature if visible in the map canvas
- Automatically pan to current feature
- Zoom to current feature

You can switch from one mode to the other by clicking the corresponding icon at the bottom right of the dialog.

You can also specify the *Default view* mode at the opening of the attribute table in *Settings* ► *Options* ► *Data Sources* menu. It can be ‚Remember last view', ‚Table view' or ‚Form view'.



12.71 Fig. : Attribute table in table view (top) vs form view (bottom)

Configuring the columns

Right-click in a column header when in table view to have access to tools that help you control:

- the *column(s) size*
- the *column(s) visibility and order*
- the *sort order of the data*

Resizing columns widths

Columns width can be set through a right-click on the column header and select either:

- *Set width…* to enter the desired value. By default, the current value is displayed in the widget
- *Set all column widths…* to the same value
- *Autosize* to resize at the best fit the column.
- *Autosize all columns*

A column size can also be changed by dragging the boundary on the right of its heading. The new size of the column is maintained for the layer, and restored at the next opening of the attribute table.

In the *Data Source Settings*, you can choose to *Autosize all columns by default when opening attribute table*, which will make „Autosize All Columns“ the default view every time attribute tables are opened in QGIS.

Hiding and organizing columns and enabling actions

By right-clicking in a column header, you can choose to *Hide column* from the attribute table (in „table view“ mode). For more advanced controls, press the Organize columns… button from the dialog toolbar or choose *Organize columns…* in a column header contextual menu.

In the new dialog, you can:

- check/uncheck columns you want to show or hide: a hidden column will disappear from every instance of the attribute table dialog until it is actively restored. It is also possible to:
  - choose *Show All* to display all the fields (columns) and actions in the table
  - choose *Hide All* to hide all the fields (columns) and actions in the table
  - use the *Toggle selection* to invert visibility of the current selection of columns. You can use *keyboard combination* for selecting multiple columns.
- drag-and-drop items to reorder the columns in the attribute table. Note that this change is for the table rendering and does not alter the fields order in the layer datasource
- add a new virtual *Actions* column that displays in each row a drop-down box or a button list of enabled actions. See *Actions Properties* for more information about actions.

Sorting rows

The rows can be sorted by any column, by clicking on the column header. A small arrow indicates the sort order (downward pointing means descending values from the top row down, upward pointing means ascending values from the top row down). You can also choose to sort the rows with the *Sort...* option of the column header context menu and write an expression. E.g. to sort the rows using multiple columns you can write `concat(col0, col1)`.

In form view, features identifier can be sorted using the *Sort by preview expression* option.

Note that sorting the rows only affects the table rendering and does not alter the features order in the layer datasource.

Patarimas

Sorting based on columns of different types

Trying to sort an attribute table based on columns of string and numeric types may lead to unexpected result because of the `concat("USE", "ID")` expression returning string values (ie, `'Borough105' < 'Borough6'`). You can workaround this by using eg `concat("USE", lpad("ID", 3, 0))` which returns `'Borough105' > 'Borough006'`.

Formatting of table cells using conditions

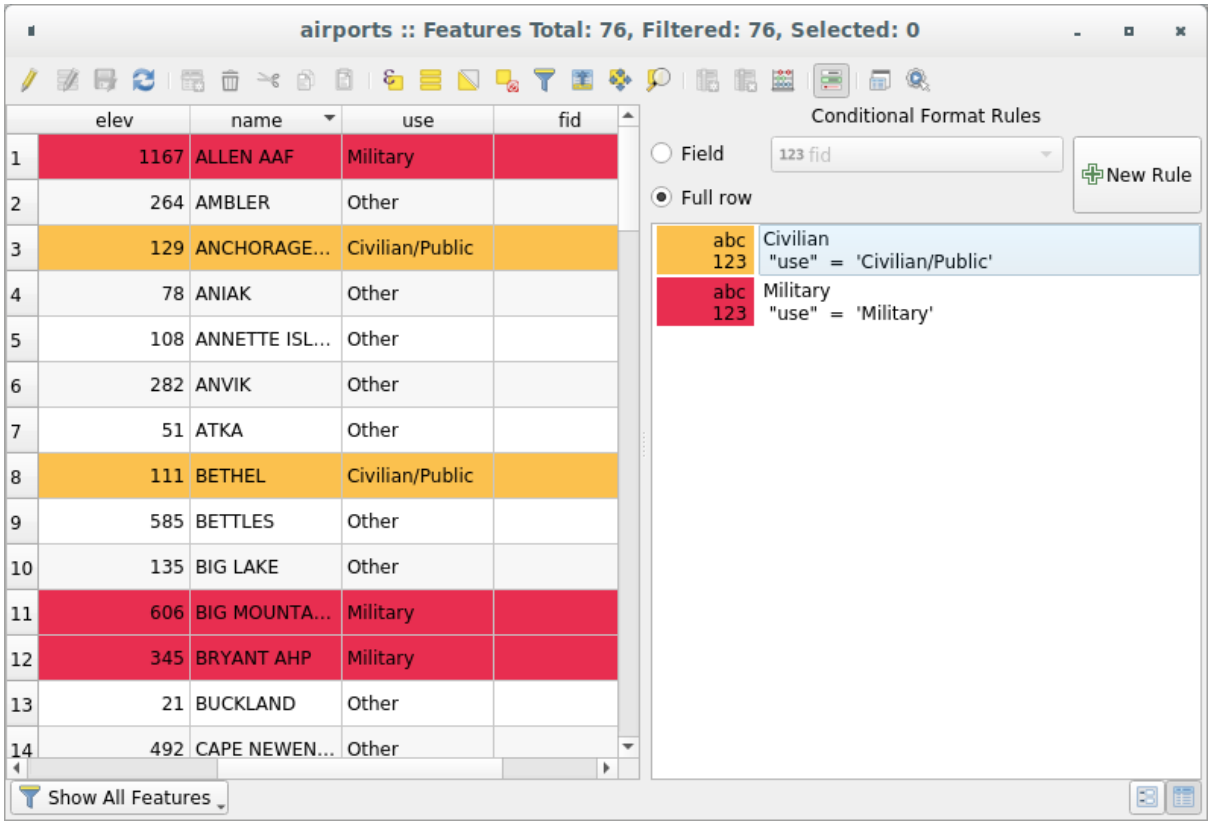
Conditional formatting settings can be used to highlight in the attribute table features you may want to put a particular focus on, using custom conditions on feature's:

- geometry (e.g., identifying multi-parts features, small area ones or in a defined map extent…);
- or field value (e.g., comparing values to a threshold, identifying empty cells, duplicates, …).

You can enable the conditional formatting panel clicking on Conditional formatting button at the top right of the attributes window in table view (not triggered in form view).

The new panel allows user to add new rules to format rendering of *Field* or *Full row*. Adding new rule opens a form to define:

- the name of the rule;
- a condition using any of the *expression builder* functions;
- the formatting: it can be choosen from a list of predefined formats or created based on properties like:
  - background and text colors;
  - use of icon;
  - bold, italic, underline, or strikeout;
  - font.



12.72 Fig. : Conditional Formatting of an attribute table

12.2.3 Interacting with features in an attribute table

Geoobjektų pažymėjimas

In table view, each row in the attribute table displays the attributes of a unique feature in the layer. Selecting a row selects the feature and likewise, selecting a feature in the map canvas (in case of geometry enabled layer) selects the row in the attribute table. If the set of features selected in the map canvas (or attribute table) is changed, then the selection is also updated in the attribute table (or map canvas) accordingly.

Rows can be selected by clicking on the row number on the left side of the row. **Multiple rows** can be marked by holding the `Ctrl` key. A **continuous selection** can be made by holding the `Shift` key and clicking on several row headers on the left side of the rows. All rows between the current cursor position and the clicked row are selected. Moving the cursor position in the attribute table, by clicking a cell in the table, does not change the row selection. Changing the selection in the main canvas does not move the cursor position in the attribute table.

In form view of the attribute table, features are by default identified in the left panel by the value of their displayed field (see *Display Properties*). This identifier can be replaced using the drop-down list at the top of the panel, either by selecting an existing field or using a custom expression. You can also choose to sort the list of features from the drop-down menu.

Click a value in the left panel to display the feature's attributes in the right one. To select a feature, you need to click inside the square symbol at the left of the identifier. By default, the symbol turns into yellow. Like in the table view, you can perform multiple feature selection using the keyboard combinations previously exposed.

Beyond selecting features with the mouse, you can perform automatic selection based on feature's attribute using tools available in the attribute table toolbar, such as (see section *Automatinis parinkimas* and subsequent for more information and use case):

- *Select By Expression…*

- *Select Features By Value…*
- *Deselect All Features from the Layer*
- *Select All Features*
- *Invert Feature Selection*.

It is also possible to *select features using forms*.

Filtering features

Once you have selected features in the attribute table, you may want to display only these records in the table. This can be easily done using the *Show Selected Features* item from the drop-down list at the bottom left of the attribute table dialog. This list offers the following filters:

- *Show All Features*
- *Show Selected Features* - same as using *Open Attribute Table (Selected Features)* from the *Layer* menu or the *Attributes Toolbar* or pressing `Shift+F6`
- *Show Features visible on map* - same as using *Open Attribute Table (Visible Features)* from the *Layer* menu or the *Attributes Toolbar* or pressing `Ctrl+F6`
- *Show Features with Failing Constraints* - features will be filtered to only show the ones which have failing *constraints*. Depending on whether the unmet constraint is hard or soft, failing field values are displayed in respectively dark or light orange cells.
- *Show Edited and New Features* - same as using *Open Attribute Table (Edited and New Features)* from the *Layer* menu or the *Attributes Toolbar*
- *Field Filter* - allows the user to filter based on value of a field: choose a column from a list, type or select a value and press `Enter` to filter. Then, only the features matching `num_field = value` or `string_field ilike '%value%'` expression are shown in the attribute table. You can check *Case sensitive* to be less permissive with strings.
- *Advanced filter (Expression)* - Opens the expression builder dialog. Within it, you can create *complex expressions* to match table rows. For example, you can filter the table using more than one field. When applied, the filter expression will show up at the bottom of the form.
- *Stored filter expressions* ►: a shortcut to *saved expressions* frequently used for filtering your attribute table.

It is also possible to *filter features using forms*.

Pastaba

Filtering records out of the attribute table does not filter features out of the layer; they are simply momentaneously hidden from the table and can be accessed from the map canvas or by removing the filter. For filters that do hide features from the layer, use the *Query Builder*.

Patarimas

Update datasource filtering with `Show Features Visible on Map`

When for performance reasons, features shown in attribute table are spatially limited to the canvas extent at its opening (see *Data Source Options* for a how-to), selecting *Show Features Visible on Map* on a new canvas extent updates the spatial restriction.

Storing filter expressions

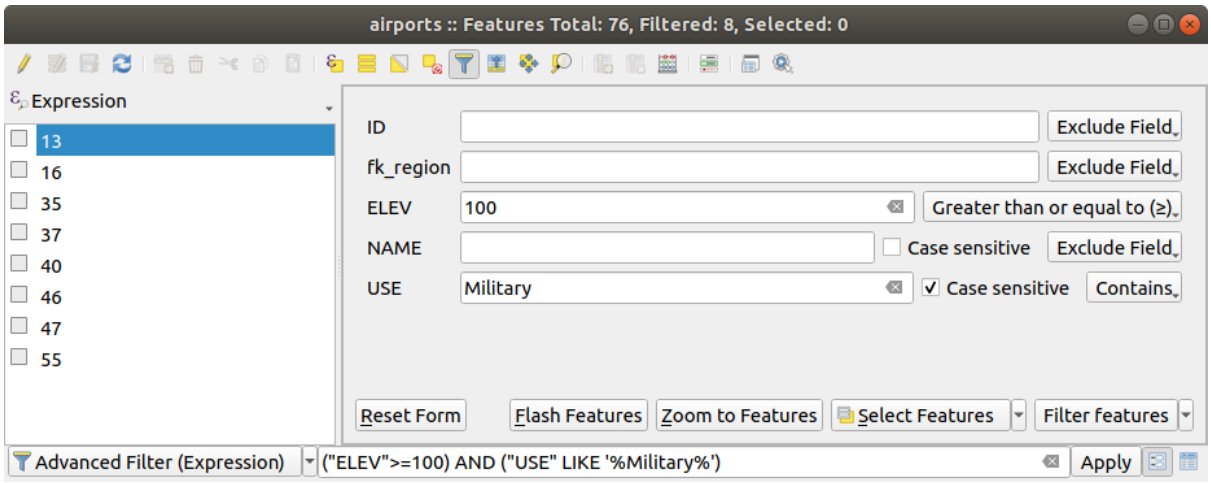
Expressions you use for attribute table filtering can be saved for further calls. When using *Field Filter* or *Advanced Filter (expression)* entries, the expression used is displayed in a text widget in the bottom of the attribute table dialog. Press the Save expression with text as name next to the box to save the expression in the project. Pressing the drop-down menu next to the button allows to save the expression with a custom name (*Save expression as...*). Once a saved expression is displayed, the button is triggered and its drop-down menu allows you to *Edit the expression* and name if any, or *Delete stored expression*.

Saved filter expressions are saved in the project and available through the *Stored filter expressions* menu of the attribute table. They are different from the *user expressions*, shared by all projects of the active user profile.

Filtering and selecting features using forms

Clicking the Filter/Select features using form or pressing `Ctrl+F` will make the attribute table dialog switch to form view and replace each widget with its search variant.

From this point onwards, this tool functionality is similar to the one described in *Parinkti geoobjektus pagal reikšmę*, where you can find descriptions of all operators and selecting modes.



12.73 Fig. : Attribute table filtered by the filter form

When selecting / filtering features from the attribute table, there is a *Filter features* button that allows defining and refining filters. Its use triggers the *Advanced filter (Expression)* option and displays the corresponding filter expression in an editable text widget at the bottom of the form.

If there are already filtered features, you can refine the filter using the drop-down list next to the *Filter features* button. The options are:

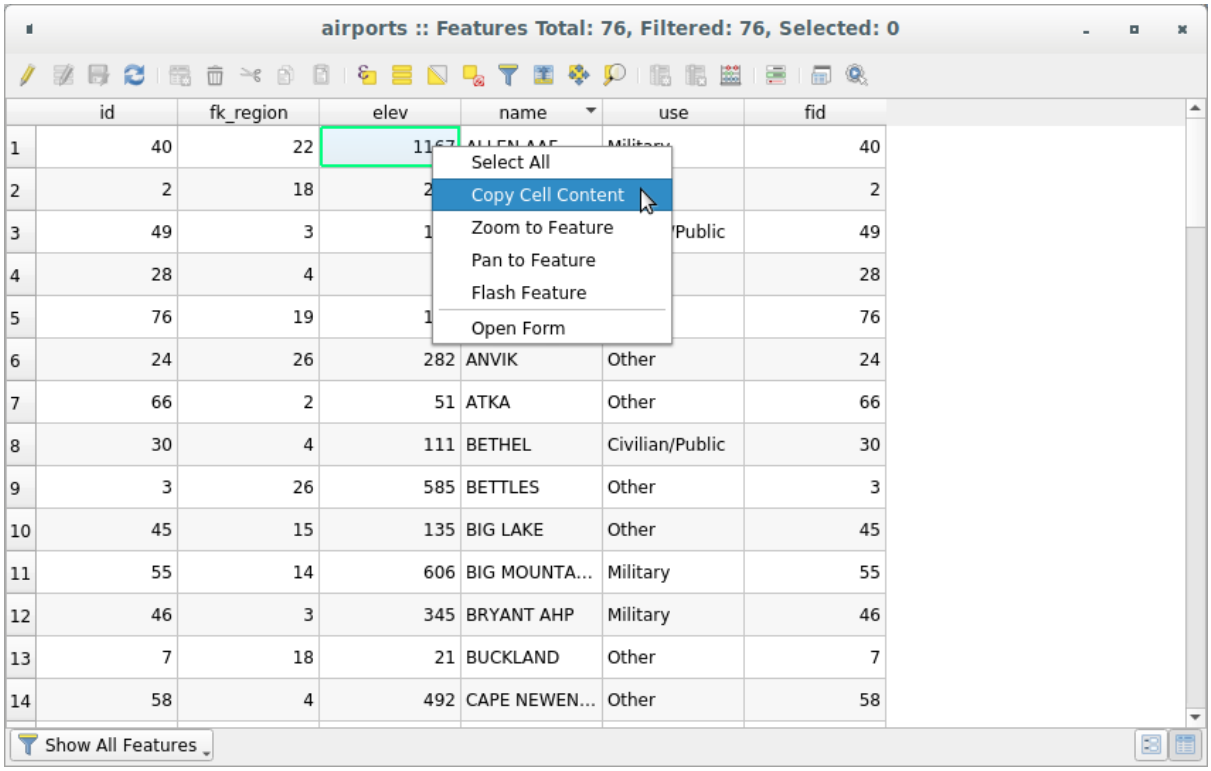
- *Filter within („AND“)*
- *Extend filter („OR“)*

To clear the filter, either select the *Show all features* option from the bottom left pull-down menu, or clear the expression and click *Apply* or press `Enter`.

More actions on features

Users have several possibilities to manipulate feature in an attribute table. Right-click in a cell and you can:

- *Select all* (`Ctrl+A`) the features;
- Copy the content of a cell in the clipboard with *Copy cell content*;
- *Zoom to feature* without having to select it beforehand;
- *Pan to feature* without having to select it beforehand;
- *Flash feature*, to highlight it in the map canvas;
- *Open form*: it toggles attribute table into form view with a focus on the clicked feature.
- Display a *list of actions*, previously enabled in the *Layer properties* ► *Actions* tab.



	id	fk_region	elev	name	use	fid
1	40	22	1167	ALLEN AAF	Military	40
2	2	18	2			2
3	49	3	1		Public	49
4	28	4				28
5	76	19	1			76
6	24	26	282	ANVIK	Other	24
7	66	2	51	ATKA	Other	66
8	30	4	111	BETHEL	Civilian/Public	30
9	3	26	585	BETTLES	Other	3
10	45	15	135	BIG LAKE	Other	45
11	55	14	606	BIG MOUNTA...	Military	55
12	46	3	345	BRYANT AHP	Military	46
13	7	18	21	BUCKLAND	Other	7
14	58	4	492	CAPE NEWEN...	Other	58

12.74 Fig. : Copy cell content button

If you want to use attribute data in external programs (such as Excel, LibreOffice, or a custom web application), select one or more row(s) and use the Copy selected rows to clipboard button or press `Ctrl+C`. Moreover, in *Settings* ► *Options* ► *Data Sources* menu you can define the format to paste to with the *Copy features as* option. More details at *Duomenų šaltinių nustatymai*.

12.2.4 Editing attribute values

In order to modify data in an attribute table, you should first toggle the layer into edit. Press the Toggle Editing button. Depending on the layer geometry type and the clipboard state, a few more tools are enabled in the attribute table top toolbar.

Editing attribute values can then be done by:

- typing the new value directly in the cell, whether the attribute table is in table or form view. Changes are hence done cell by cell, feature by feature;
- using the *field calculator*: update in a row a field that may already exist or to be created but for multiple features. It can be used to create virtual fields;
- using the quick field *calculation bar*: same as above but for only existing field;
- or using the *multi edit* mode: update in a row multiple fields for multiple features.

Putting the layer into edit mode will also allow you to Paste features from clipboard (`Ctrl+V`) Cut selected rows to clipboard (`Ctrl+X`) or Delete selected features. More details at *Cutting, Copying and Pasting Features*.

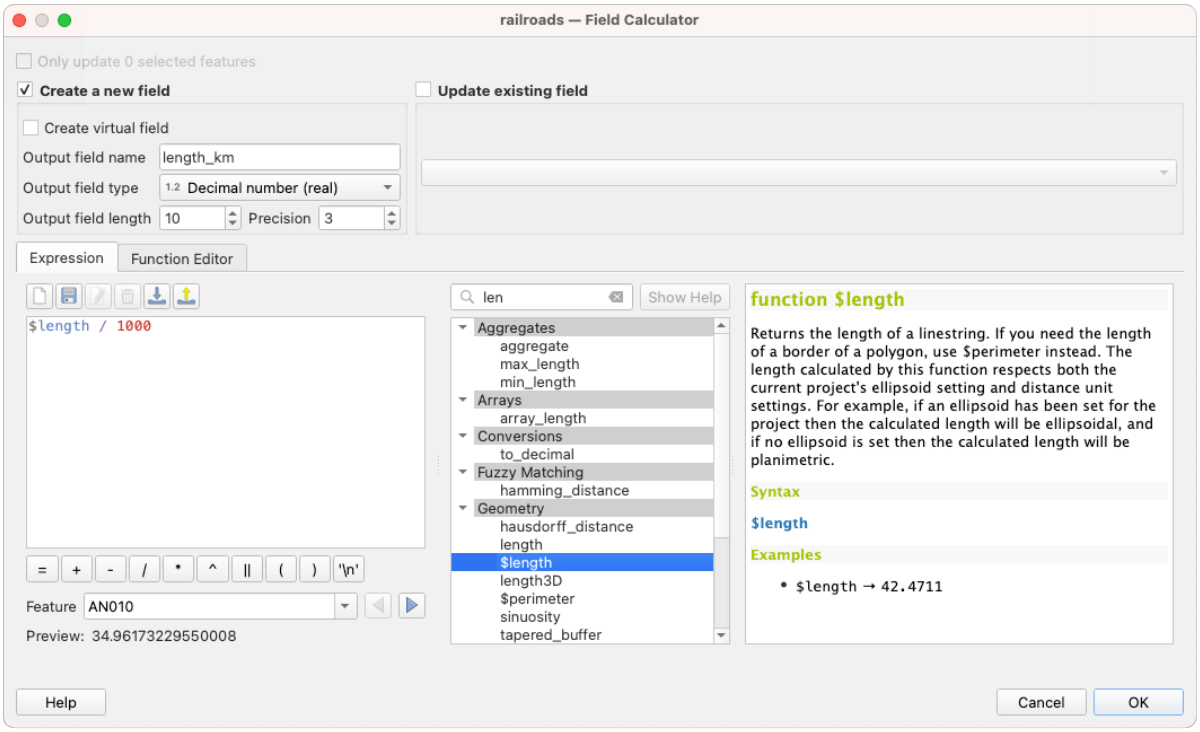
Using the Field Calculator

The Field Calculator button in the attribute table allows you to perform calculations on the basis of existing attribute values or defined functions, for instance, to calculate length or area of geometry features. The results can be used to update an existing field, or written to a new field (that can be a *virtual* one).

The field calculator is available on any layer that supports edit. When you click on the field calculator icon the dialog opens (see Fig. 12.75). If the layer is not in edit mode, a warning is displayed and using the field calculator will cause the layer to be put in edit mode before the calculation is made.

Based on the *Expression Builder* dialog, the field calculator dialog offers a complete interface to define an expression and apply it to an existing or a newly created field. To use the field calculator dialog, you must select whether you want to:

1. apply calculation on the whole layer or on selected features only
2. create a new field for the calculation or update an existing one.



12.75 Fig. : Field Calculator

If you choose to add a new field, you need to enter a field name, a field type (integer, real, date or string) and if needed, the total field length and the field precision. For example, if you choose a field length of 10 and a field precision of 3, it means you have 7 digits before the dot, and 3 digits for the decimal part.

A short example illustrates how field calculator works when using the *Expression* tab. We want to calculate the length in km of the `railroads` layer from the QGIS sample dataset:

1. Load the shapefile `railroads.shp` in QGIS and press Open Attribute Table.
2. Click on Toggle editing mode and open the Field Calculator dialog.
3. Select the *Create a new field* checkbox to save the calculations into a new field.
4. Set *Output field name* to `length_km`
5. Select `Decimal number (real)` as *Output field type*
6. Set the *Output field length* to `10` and the *Precision* to `3`
7. Double click on `$length` in the *Geometry* group to add the length of the geometry into the Field calculator expression box (you will begin to see a preview of the output, up to 60 characters, below the expression box updating in real-time as the expression is assembled).
8. Complete the expression by typing `/ 1000` in the Field calculator expression box and click *OK*.
9. You can now find a new *length_km* field in the attribute table.

Creating a Virtual Field

A virtual field is a field based on an expression calculated on the fly, meaning that its value is automatically updated as soon as an underlying parameter changes. The expression applies to all the features in the layer and is set once; you no longer need to recalculate the field each time underlying values change. For example, you may want to use a virtual field if you need area to be evaluated as you digitize features or to automatically calculate a duration between dates that may change (e.g., using `now()` function).

Creating a virtual field is done through the *Field calculator* dialog and follows the *same procedure* as for regular fields. Simply remember to check the *Create virtual field* option and use a field type compatible with the data your expression would generate.

Editing a virtual field is done through the *Fields* tab of the layer properties dialog (see *Fields Properties*). The expression defining the field is exposed in the *Comment* column, and pressing the button next to it opens an expression editor window for update.

Pastaba

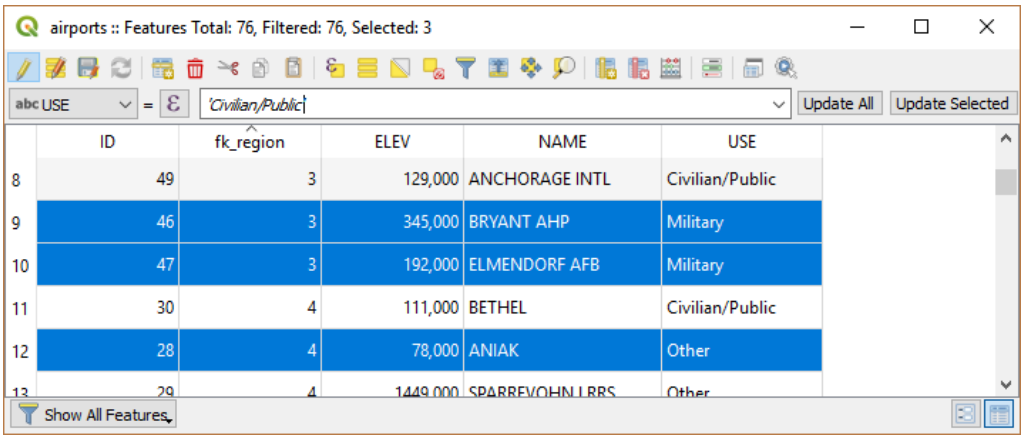
Use of Virtual Fields

- A field can be set virtual only at its creation.
- Virtual fields are not permanent in the layer attributes, meaning that they're only saved and available in the project file they've been created.

Using the Quick Field Calculation Bar

While Field calculator is always available, the quick field calculation bar on top of the attribute table is only visible if the layer is in edit mode. Thanks to the expression engine, it offers a quicker access to edit an already existing field:

1. Select the field to update in the drop-down list.
2. Fill the textbox with a value, an expression you directly write or build using the expression button.
3. Click on *Update All*, *Update Selected* or *Update Filtered* button according to your need.



12.76 Fig. : Quick Field Calculation Bar

Editing multiple fields

Unlike the previous tools, multi edit mode allows multiple attributes of different features to be edited simultaneously. When the layer is toggled to edit, multi edit capabilities are accessible:

- using the Toggle multi edit mode button from the toolbar inside the attribute table dialog;
- or selecting *Edit* ► *Modify attributes of selected features* menu.

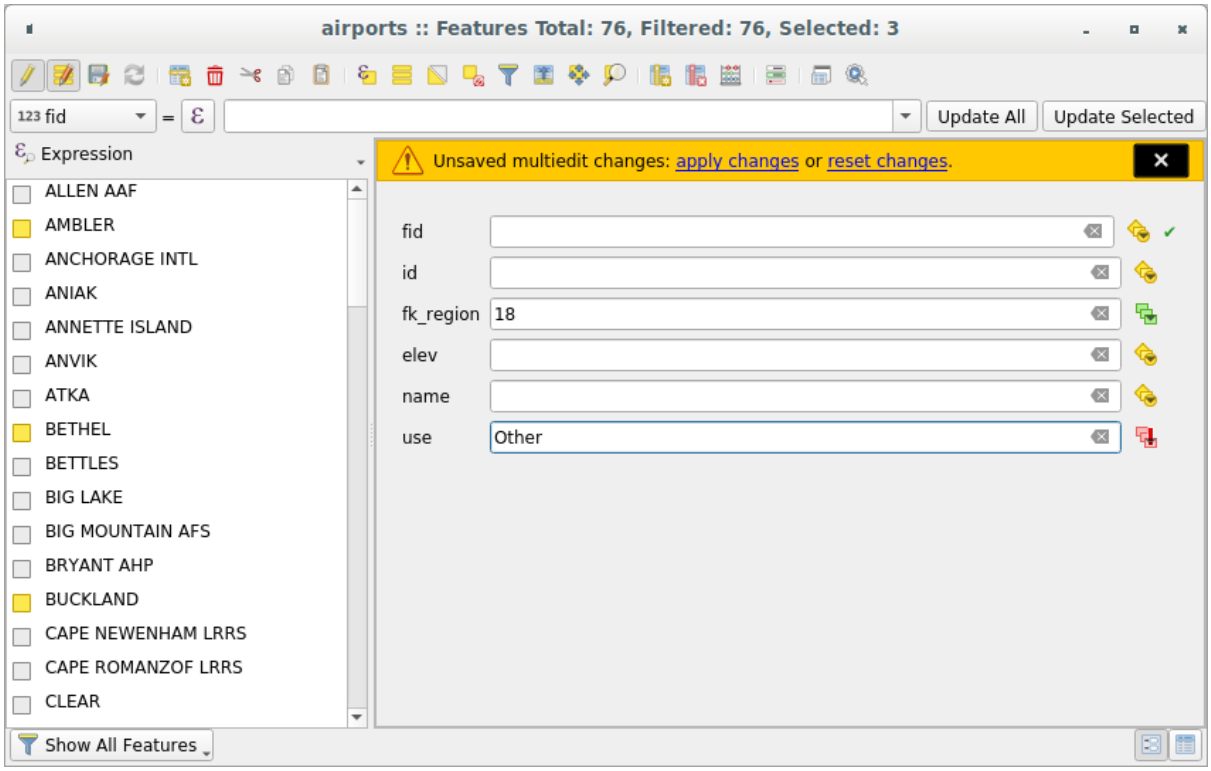
Pastaba

Unlike the tool from the attribute table, hitting the *Edit* ► *Modify Attributes of Selected Features* option provides you with a modal dialog to fill attributes changes. Hence, features selection is required before execution.

In order to edit multiple fields in a row:

1. Select the features you want to edit.
2. From the attribute table toolbar, click the button. This will toggle the dialog to its form view. Feature selection could also be made at this step.
3. At the right side of the attribute table, fields (and values) of selected features are shown. New widgets appear next to each field allowing for display of the current multi edit state:
   - The field contains different values for selected features. It's shown empty and each feature will keep its original value. You can reset the value of the field from the drop-down list of the widget.
   - All selected features have the same value for this field and the value displayed in the form will be kept.
   - The field has been edited and the entered value will be applied to all the selected features. A message appears at the top of the dialog, inviting you to either apply or reset your modification.

   Clicking any of these widgets allows you to either set the current value for the field or reset to original value, meaning that you can roll back changes on a field-by-field basis.



12.77 Fig. : Editing fields of multiple features

4. Make the changes to the fields you want.
5. Click on **Apply changes** in the upper message text or any other feature in the left panel.

Changes will apply to **all selected features**. If no feature is selected, the whole table is updated with your changes. Modifications are made as a single edit command. So pressing Undo will rollback the attribute changes for all selected features at once.

Pastaba

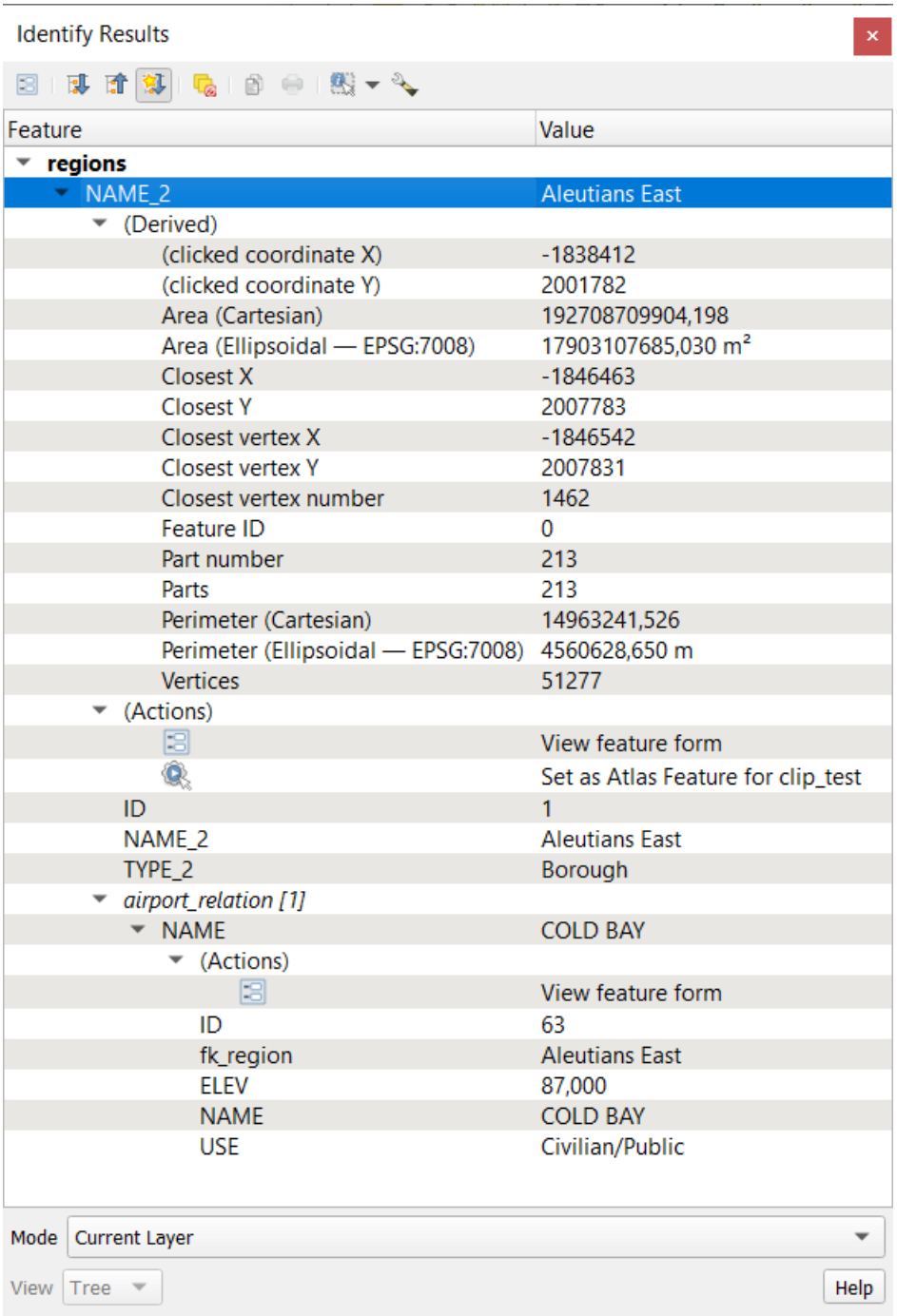
Multi edit mode is only available for auto generated and drag and drop forms (see *Customizing a form for your data*); it is not supported by custom ui forms.

12.2.5 Exploring features attributes through the Identify Tool

The *Identify features* tool can be used to display all attributes of a feature in the map canvas. It is a quick way to view and verify all data without having to search for it in the attribute table.

To use the *Identify features* tool for vector layers, follow these steps:

1. Select the vector layer in the Layers panel.
2. Click on the *Identify features* tool in the toolbar or press `Ctrl+Shift+I`.
3. Click on a feature in the map view.



12.78 Fig. : Identify Results dialog

The *Identify results* panel will display different features information in a tree view. There are two columns in the panel: *Feature* and *Value*. The first item is the name of the layer and its children are its identified feature(s). Each feature is identified by the *display name* field along with its value. Other information about the feature follows:

- **Derived** section - those are the information calculated or derived from other information in the layer. For example, the area of a polygon or the length of a line. General information that can be found in this section:
  - Depending on the geometry type, the cartesian measurements of length, perimeter or area in the layer's CRS units. For 3D line vectors, the cartesian line length is available.
  - Depending on the geometry type and if an ellipsoid is set in the *Project Properties* dialog (*General* ► *Measurements*), ellipsoidal values of length, perimeter, or area using the specified units.

  - The count of geometry parts in the feature and the number of the part clicked.
  - The count of vertices in the feature.
  - Coordinate information, using the project properties *Coordinates display* settings:
    * `X` and `Y` coordinate values of the clicked point
    * the number of the closest vertex to the clicked point
    * `X` and `Y` coordinate values of the closest vertex (and `Z`/`M` if applicable)
    * if you click on a curved segment, the radius of that section is also displayed.
    * if both the vector layer and the project have vertical datums set and they differ, the `Z` value will be displayed for both datums.

- **Actions**: Actions can be added to the identify feature windows. The action is run by clicking on the action label. By default, only one action is added, namely `View feature form` for editing. You can define more actions in the layer's properties dialog (see *Actions Properties*).
- **Duomenų atributai**: Tai sąrašas paspausto geoobjekto atributų laukų ir jų reikšmių.

  Pastaba

  Links in the feature's attributes are clickable from the *Identify Results* panel and will open in your default web browser.

- When a vector layer has defined *relations*, the *Identify Results* panel can display both **referenced** and **referencing** related features. To view these relations, ensure that the *Show relations* option is enabled in the *Identify Settings*. Available information for each related feature include:
  - the name of the relation and the linked layer
  - the entry in the referenced or referencing field, identifying the feature
  - the activated actions
  - the list of attribute fields and values

  You can expand related features to explore connected records through multiple levels, including many-to-many (n:m) and polymorphic relations. Only nodes you explicitly expand are loaded, preventing excessive nesting.

  To focus on a specific related record, right-click it and choose *Identify Feature*. This re-centers the identify results on the selected feature, starting a new identification tree from it and effectively limiting the visible nesting depth. Features already shown in ancestor nodes are automatically omitted to avoid duplicates or circular relations.

12.3 Redagavimas

QGIS has various capabilities for editing OGR, SpatiaLite, PostgreSQL, MS SQL Server and Oracle Spatial vector layers and tables. They can be of 2D or 3D geometry type.

Pastaba

GRASS sluoksnių redagavimo procedūra šiek tiek skiriasi - daugiau informacijos rasite skyriuje *Digitizing and editing a GRASS vector layer*.

Dėmesio

Redagavimai vienu metu

QGIS neseka, ar kas nors kitas redaguoja tą patį geoobjektą tuo pačiu metu, kaip ir jūs. Laimi paskutinis įrašęs pakeitimus.

Patarimas

Keitimų tikrinimas

Galima įjungti tęstinį sluoksnio tikrinimą kortelėje *Sluoksnio savybės* ► *Skaitmeninimas*. Daugiau informacijos rasite *Digitizing Properties*.

12.3.1 Setting the snapping tolerance and search radius

Under the *Settings* ► *Options…* ► *Digitizing* menu, QGIS provides a number of parameters to configure default behaviour of editing tools. More information at *Skaitmeninimo nustatymai*.

For optimal and accurate editing of vector layer geometries, we need to set an appropriate value of snapping tolerance and search radius for features vertices. The *Snapping* group provides related options, namely handling of the snapping tolerance and the search radius.

- Snapping tolerance: When you add a new vertex or move an existing one, the snapping tolerance is the distance QGIS uses to search for the closest vertex or segment you are trying to connect to. If you are not within the snapping tolerance, QGIS will leave the vertex where you release the mouse button, instead of snapping it to an existing vertex or segment.

  The tolerance setting affects all tools that work with snapping and applies by default to new layers and projets. It can however be overridden at layer level (see *Snapping and Digitizing Options*).

- Search radius: *Search radius for vertex edits* is the distance QGIS uses to `search` for the vertex to select when you click on the map. If you are not within the search radius, QGIS will not find and select any vertex for editing.

Snap tolerance and search radius are set in `map units` or `pixels`. You may need to experiment to get them right. If you specify a too big tolerance, QGIS may snap to the wrong vertex, especially if you are dealing with a large number of vertices in close proximity. The smaller the search radius, the more difficult it will be to hit what you want to move.

12.3.2 Snapping and Digitizing Options

Global *snapping and digitizing settings* (snapping mode, tolerance value, and units…) can be overridden in the project from the *Project* ► *Snapping Options…* menu. In the *Snapping and Digitizing Options*, you can also configure some other properties (snapping layers, scale limit, topology…) The *Snapping Toolbar* gives access to most of these features.

By default, snapping is disabled in a project until you press the Enable snapping button or press `S`. The snapping mode, tolerance value, and units can also be configured in this toolbar.

Snapping properties

There are three options to select the layer(s) to snap to:

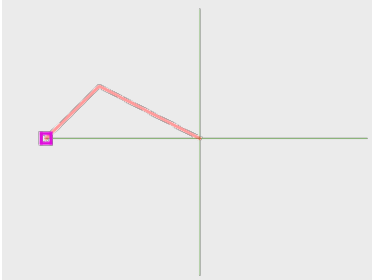
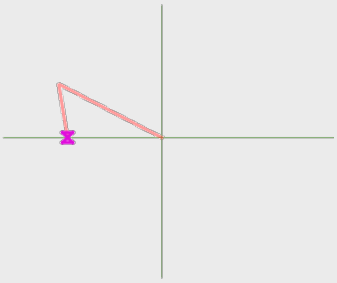
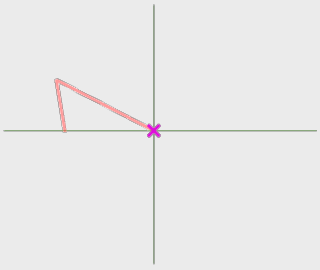
- *All layers*: quick setting for all visible layers in the project so that the pointer snaps to all vertices and/or segments. In most cases, it is sufficient to use this snapping mode, but beware when using it for projects with many vector layers, as it may affect performance.
- *Current layer*: only the active layer is used, a convenient way to ensure topological consistency within the layer being edited.
- *Advanced Configuration*: allows you to enable and adjust snapping mode, tolerance and units, overlaps and scales of snapping on a layer basis (see Fig. 12.79). If you need to edit a layer and snap its vertices to another, make sure that the target layer is checked and increase the snapping tolerance to a higher value. Snapping will not occur to a layer that is not checked in the snapping options dialog.

When moving or creating vertex, you can opt for the following snapping modes:

- *Vertex*
- *Segment*: snaps along a line or a polygon perimeter. If topological editing is enabled, then a new vertex is added at the snapping location.
- *Area*: guarantees that the snap point lies anywhere on a polygon's area, not necessarily on its boundary
- *Centroid*: snaps to the centroid of the geometry of a feature. In case of a multipart geometry, the target point may be distinct from the existing parts.
- *Middle of Segments* on line or polygon feature
- *Line Endpoints*: snaps to the first or last vertex of every part of a line or polygon feature.

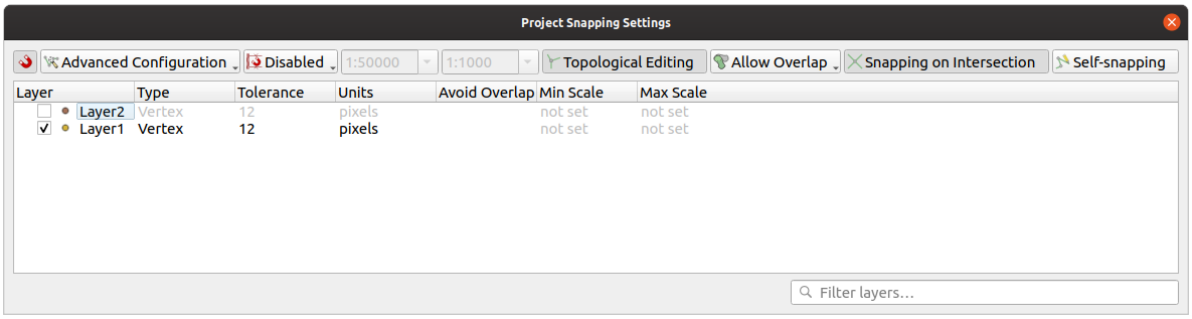
QGIS will show different *snap* icons depending on the kind of *snap*:

12.2 Table : Snapping icons

		
Snapping to a vertex: box icon	Snapping to a segment: hourglass icon	Snapping to an intersection: cross icon

Note that it is possible to change the color of these icons in the *Digitizing* part of the global settings.

The tolerance values can be set either in the project's `map units` or in `pixels`. The advantage of choosing `pixels` is that it keeps the snapping constant at different map scales. 10 to 12 pixels is normally a good value, but it depends on the DPI of your screen. Using map units allows the tolerance to be related to real ground distances. For example, if you have a minimum distance between elements, this option can be useful to ensure that you don't add vertices too close to each other.



12.79 Fig. : Snapping options (Advanced Configuration mode)

Pastaba

By default, only visible features (the features whose style is displayed, except for layers where the symbology is „No symbols“) can be snapped. You can enable the snapping on invisible features by checking *Enable snapping on invisible features* under the *Settings* ► *Options* ► *Digitizing* tab.

Patarimas

Enable snapping by default

You can set snapping to be enabled by default on all new projects in the *Settings* ► *Options* ► *Digitizing* tab. You can also set the default snapping mode, tolerance value, and units, which will populate the *Snapping Options* dialog.

Enable snapping on intersections

Another available option is to use *snapping on intersection*, which allows you to snap to geometry intersections of snapping enabled layers, even if there are no vertices at the intersections.

Limit snapping to a scale range

In some cases snapping can become very slow. This is often caused by the amount of features in some layers that require a heavy index to compute and maintain. Some parameters exist to enable snapping only when the map view is inside a relevant scale range. This allows to only do the costly index computation related to snapping at a scale where drawing is relevant.

Scale limit to snapping is configured in *Project* ► *Snapping Options…*. Limiting snapping to scale is only available in *Advanced Configuration* mode.

To limit snapping to a scale range you have three modes available:

- *Disabled*: Snapping is enabled whatever the current map scale is. This is the default mode.
- *Global*: Snapping is limited and only enabled when the current scale of the map is between a global minimum and a global maximum value. When selecting this mode two widgets become available to configure the range of scales in which snapping is enabled.
- *Per layer*: The snapping scale range limit is defined for each layer. When selecting this mode two columns become available to configure the minimum and maximum scales for each layer.

Please note that the minimum and maximum scales follow the QGIS convention: minimum scale is the most „zoomed out“ scale while maximum scale is the most „zoomed in“. A minimum or maximum scale that is set to „0“ or „not set“ is considered not limiting.

Self-snapping

The Self-snapping option allows you to snap to the geometry that is being edited. Combined with the *advanced digitizing panel*, this provides a handy way to digitize new edges relative to the previous edges or vertices. Self-snapping can cause invalid geometries, use with caution.

Snapping on custom grid

A snapping distance can also be customized on a layer basis in the *Digitizing* tab of the layer properties dialog. With setting the *Geometry precision* distance, you enable a dotted grid visible when the map canvas is at a coherent scale for display. Snapping can then be performed on the dots of the grid: an added or modified geometry will have all of its vertices snapped automatically to the closest node of the grid. More information at *Digitizing Properties*.

12.3.3 Topological editing

In addition to these snapping options, the *Snapping options...* dialog (*Project* ► *Snapping options*) and the *Snapping* toolbar allow you to enable / disable some other topological functionalities.

Enable topological editing

The Topological editing button helps when editing and maintaining features with common boundaries. With this option enabled, QGIS ‚detects‘ shared boundaries. When you move common vertices/segments, QGIS will also move them in the geometries of the neighboring features.

Topological editing works with features from different layers, as long as the layers are visible and in editing mode.

In layer with Z or M values, topological editing will interpolate the Z or M value of the vertex based on the value of the edge used for the connection.

Overlapping control

Overlapping prevents you from drawing new features that overlap existing ones in the selected layer, speeding up digitizing of adjacent polygons. It can be controlled by the overlap tool. Three modes are available:

1. *Allow Overlap* (default)
2. *Avoid Overlap on Active Layer*: prevents any overlap with other features from the layer being edited. Digitize the new geometries so that they overlap their neighbours and QGIS will cut the overlapping part(s) of the new geometries and snap them to the boundary of the existing features. The advantage is that you don't have to digitize the common vertices on boundary.
3. *Follow Advanced Configuration*: allows the overlapping setting to be set on a layer basis in the *Advanced configuration* view mode.

> **Pastaba**
>
> If the new geometry is totally covered by existing ones, it gets cleared, and QGIS will show an error message.

Įspėjimas

Use cautiously the *Avoid overlap* option

Since this option will cut new overlapping geometries of any polygon layer, you can get unexpected geometries if you forget to uncheck it when no longer needed.

Automatic Tracing

Usually, when using capturing map tools (add feature, add part, add ring, reshape and split), you need to click each vertex of the feature. With the automatic tracing mode, you can speed up the digitization process as you no longer need to manually place all the vertices during digitization:

1. Enable the Tracing tool (in the *Snapping* toolbar) by pushing the icon or pressing `T` key.
2. *Snap to* a vertex or segment of a feature you want to trace along.
3. Move the mouse over another vertex or segment you'd like to snap and, instead of the usual straight line, the digitizing rubber band represents a path from the last point you snapped to the current position. The tool also works with curved geometries.

   QGIS actually uses the underlying features topology to build the shortest path between the two points. Tracing requires snapping to be activated in traceable layers to build the path. You should also snap to an existing vertex or segment while digitizing and ensure that the two nodes are topologically connectable through existing features edges, otherwise QGIS is unable to connect them and thus traces a single straight line.
4. Click and QGIS places the intermediate vertices following the displayed path.

Unfold the Enable Tracing icon and set the *Offset* option to digitize a path parallel to the features instead of tracing along them. A positive value shifts the new drawing to the left side of the tracing direction and a negative value does the opposite.

Pastaba

Adjust map scale or snapping settings for an optimal tracing

If there are too many features in map display, tracing is disabled to avoid potentially long tracing structure preparation and large memory overhead. After zooming in or disabling some layers the tracing is enabled again.

Pastaba

Does not add topological points

This tool does not add points to existing polygon geometries even if *Topological editing* is enabled. If geometry precision is activated on the edited layer, the resulting geometry might not exactly follow an existing geometry.

Patarimas

Quickly enable or disable automatic tracing by pressing the `T` key

By pressing the `T` key, tracing can be enabled/disabled anytime (even while digitizing a feature), so it is possible to digitize parts of the feature with tracing enabled and other parts with tracing disabled. Tools behave as usual when tracing is disabled.

Patarimas

Convert tracing to curved geometries

By using *Settings* ► *Options* ► *Digitizing* ► *Tracing* you can create curved geometries while digitizing. See *digitizing options*.

12.3.4 Digitizing an existing layer

By default, QGIS loads layers read-only. This is a safeguard to avoid accidentally editing a layer if there is a slip of the mouse. However, you can choose to edit any layer as long as the data provider supports it (see *Exploring Data Formats and Fields*), and the underlying data source is writable (i.e., its files are not read-only).

Patarimas

Restrict edit permission on layers within a project

From the *Project* ► *Properties...* ► *Data Sources* ► *Layers Capabilities* table, you can choose to set any layer read-only regardless the provider permission. This can be a handy way, in a multi-users environment to avoid unauthorized users to mistakenly edit layers (e.g., Shapefile), hence potentially corrupt data. Note that this setting only applies inside the current project.

In general, tools for editing vector layers are divided into a digitizing and an advanced digitizing toolbar, described in section *Advanced digitizing*. You can select and unselect both under *View* ► *Toolbars* ►.

Using the basic digitizing tools, you can perform the following functions:

12.3 Table : Vector layer basic editing toolbar

Įrankis	Tikslas	Įrank	Tikslas
	Access to save, rollback or cancel changes in all or selected layers simultaneously		Turn on or off edit status of selected layer(s) based on the active layer status
	Save edits to the active layer		
	Digitize using straight segments		Digitize using curve lines
	Enable freehand digitizing		Digitize polygon of regular shape
	Add new record		Add Feature: Capture Point
	Add Feature: Capture Line		Add Feature: Capture Polygon
	Vertex Tool (All Layers)		Vertex Tool (Current Layer)
Show Vertex Editor	Set whether the vertex editor panel should auto-open		Modify the attributes of all selected features simultaneously
	Delete Selected features from the active layer		Cut Features from the active layer
	Copy selected Features from the active layer		Paste Features into the active layer
	Undo changes in the active layer		Redo changes in active layer

Note that while using any of the digitizing tools, you can still *zoom or pan* in the map canvas without losing the focus on the tool.

All editing sessions start by choosing the Toggle editing option found in the context menu of a given layer, from the attribute table dialog, the digitizing toolbar or the *Layer* menu.

Once the layer is in edit mode, additional tool buttons on the editing toolbar will become available and markers will appear at the vertices of all features unless *Show markers only for selected features* option under *Settings* ► *Options…* ► *Digitizing* menu is checked.

Patarimas

Save Regularly

Remember to Save Layer Edits regularly. This will also check that your data source can accept all the changes.

Geometry editing techniques

When a geometry drawing tool (mainly the ones that add, split, reshape features) is enabled for a line or polygon based layer, you can select the technique for adding new vertices:

- The Digitize with Segment: draws straight segment whose start and end points are defined by left clicks.
- The Digitize with Curve: draws curve line based on three consecutive nodes defined by left clicks (start, point along the arc, end). If the geometry type does not support curves, then consecutive smaller segments are used to approximate the curvature.
- The Stream Digitizing: draws lines in freehand mode, i.e. nodes are added following cursor movement in the map canvas and a *Streaming Tolerance*. The streaming tolerance defines the spacing between consecutive vertices. Currently, the only supported unit is pixels (`px`). Only the starting left click and the ending right click are necessary in this mode.
- The Digitize Shape: triggers tools on the *Shape Digitizing Toolbar* to draw a polygon of a regular shape.

The selected technique remains while switching among the digitizing tools. You can combine any of the first three methods while drawing the same geometry.

Adding Features

Depending on the layer type, you can use the Add Record, Add Point Feature, Add Line Feature or Add Polygon Feature icons on the toolbar to add new features into the current layer.

To add a geometryless feature, click on the Add Record button and you can enter attributes in the feature form that opens.

To create features with the spatially enabled tools, you first digitize the geometry then enter its attributes. To digitize the geometry:

1. (Optional as it is the default) Select the Digitize With Segment geometry drawing method
2. Left-click on the map area to create the first point of your new feature. For point features, this should be enough and trigger, if required, the feature form to fill in their attributes.
3. For line or polygon geometries, keep on left-clicking for each additional point you wish to capture. You can rely on the *snapping to features* options, the *snap-to-grid* or the *advanced digitizing* panel to accurately position each vertex.

   Along with drawing straight segments between nodes you click one by one, lines and polygons can be:

   - *traced automatically*, accelerating the digitization. This will create consecutive straight lines between the vertices you place, following existing features.
   - free-hand digitized, pressing `R` or activating Stream Digitizing.

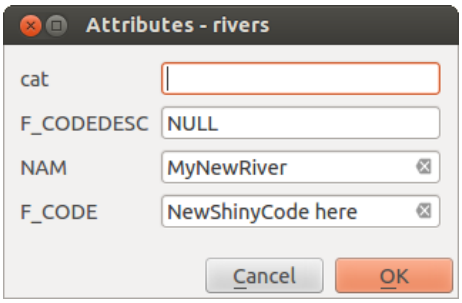
- drawn as curve, pressing `Ctrl+Shift+G` or activating Digitize with Curve.

> **Pastaba**
>
> While digitizing line or polygon geometries, you can switch back and forth between the geometry drawing methods, allowing you to create features mixing straight segments, free-hand ones and curved parts.

4. Press `Delete` or `Backspace` key to revert the last node(s) you may wrongly add.
5. When you have finished adding points, right-click anywhere on the map area to confirm you have finished entering the geometry of that feature.

> **Patarimas**
>
> **Customize the digitizing rubber band**
>
> While capturing polygon, the by-default red rubber band can hide underlying features or places you'd like to capture a point. This can be fixed by setting a lower opacity (or alpha channel) to the rubber band's *Fill Color* in *Settings* ► *Options* ► *Digitizing* menu. You can also avoid the use of the rubber band by checking *Don't update rubber band during node editing*.

6. For line feature pressing `Shift` + right-click will close the line automatically.
7. The attribute window will appear, allowing you to enter the information for the new feature. Fig. 12.80 shows setting attributes for a fictitious new river. However, in the *Digitizing* menu under the *Settings* ► *Options* menu, you can also:
   - *Suppress attributes pop-up windows after each created feature* to avoid the form opening;
   - or *Reuse last entered attribute values* to have fields automatically filled at the opening of the form and just have to type changing values.



12.80 Fig. : Enter Attribute Values Dialog after digitizing a new vector feature

Vertex tool

QGIS provides two tools to interact with vector features vertices:

- Vertex Tool (Current Layer): only overlaid features in the active layer (in the *Layers* panel) are affected
- Vertex Tool (All Layers): any overlaid features in all editable layers are affected. This allows you to edit features without switching the active layer or edit multiple layers at once (e.g., country and their regions boundaries)

For any editable vector layer, the vertex tools provide manipulation capabilities of feature vertices similar to CAD programs. It is possible to select multiple vertices at once and to move, add or delete them altogether. The vertex tools

also support the topological editing feature. They are selection persistent, so when some operation is done, selection stays active for this feature and tool.

> **Patarimas**
>
> **Drawing a series of new vertices**
>
> The vertex tool does not support *automatic tracing* and is optimized for editing individual vertices and moving or deleting an arbitrary selection of multiple vertices. Try using the Reshape Features tool when you need to replace or insert a series of new vertices, especially if you want to trace other features.

It is important to set the property *Settings* ► *Options* ► *Digitizing* ► *Search Radius:* 1,00 to a number greater than zero. Otherwise, QGIS will not be able to tell which vertex is being edited and will display a warning.

> **Patarimas**
>
> **Vertex Markers**
>
> QGIS supports different kinds of vertex markers: ‚Semi-transparent circle', ‚Cross' and ‚None'. To change the marker style, choose *Options* from the *Settings* menu, click on the *Digitizing* tab and select the appropriate entry.

Basic operations

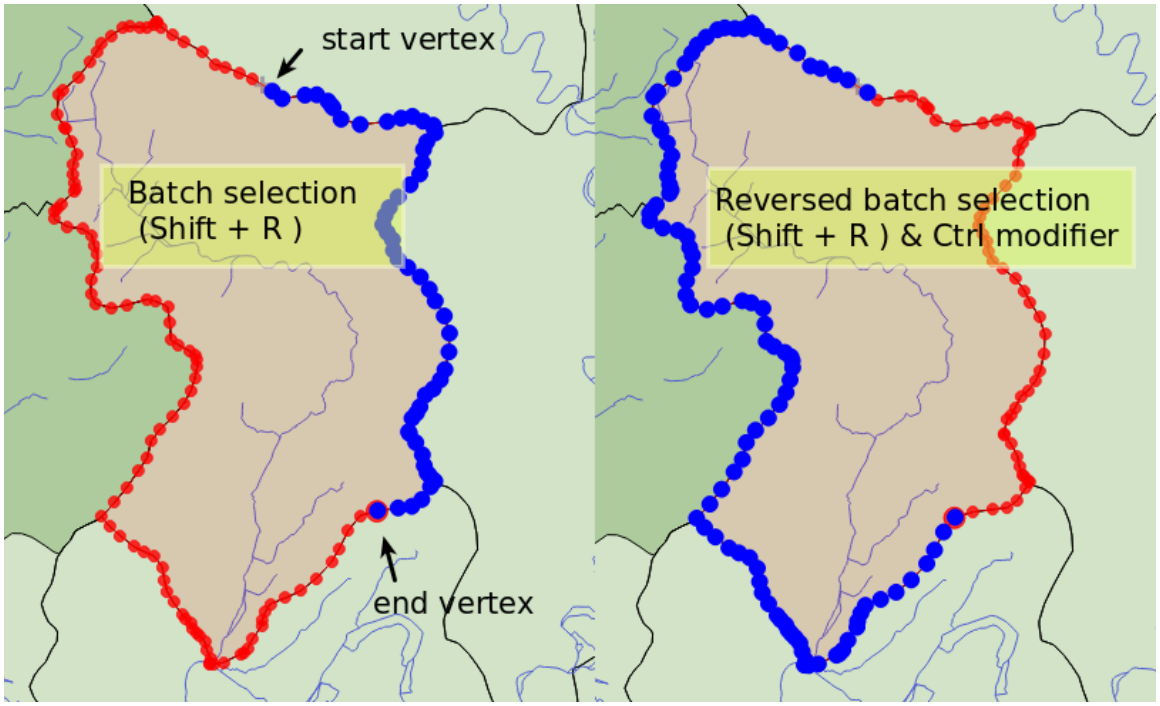
Given a layer in edit mode, start by activating the vertex tool. Red circles will appear when hovering vertices.

- **Selecting vertices**: You can select vertices by:
  - Clicking on them one at a time holding `Shift` key pressed
  - Click-and-dragging a rectangle surrounding the target vertices
  - Drawing a polygon surrounding the target vertices: Hold `Alt` and click using the vertex tool to start digitizing a polygon. Each subsequent click adds a new vertex to the rubberband polygon. `Backspace` or `Delete` removes last added rubberband vertex. `Esc` cancels the polygon selection mode, as also does backspacing/deleting all of the rubberband's vertices. Right click finalizes the polygon digitizing and selects all vertices within the rubberband polygon.

  When a vertex is selected, its color changes to blue. To add more vertices to the current selection, hold down the `Shift` key while proceeding as above. To remove vertices from the selection, hold down `Ctrl`.

  > **Patarimas**
  >
  > **Feature selection bounds vertex tool**
  >
  > Vertices can be selected accross different features (or layers). If you are looking for vertices of a specific feature in a crowded place, first select that feature. Then draw the rectangle or polygon selector with the vertex tool around the vertices: only the selected feature's vertices are selected.
  >
  > This is also the case if you display the feature in the *vertex editor* panel.

- **Batch vertex selection mode**: The batch selection mode can be activated by pressing `Shift+R`. Select a first node with one single click, and then hover **without clicking** another vertex. This will dynamically select all the nodes in between using the shortest path (for polygons).

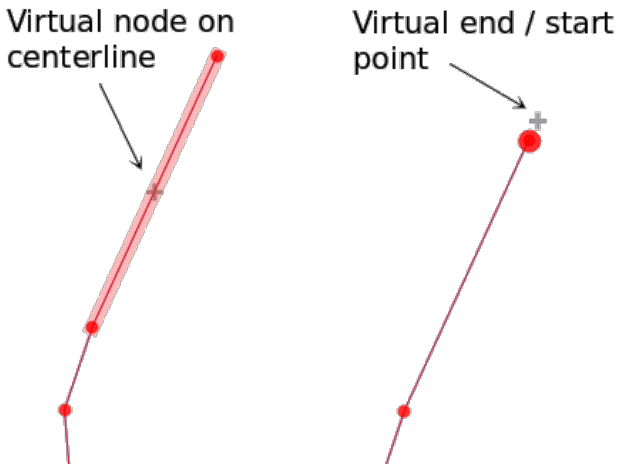


12.81 Fig. : Batch vertex selection using `Shift+R`

Press `Ctrl` will invert the selection, selecting the longest path along the feature boundary. Ending your node selection with a second click, or pressing `Esc` will escape the batch mode.

- **Adding vertices**: To add a vertex to a line or polygon geometry, hold `Shift` and double-click the place on the segment.

  When hovering a segment, a virtual new node appears on the center. Click on it, move the cursor to a new location and click again to add a new vertex. For lines, a virtual node is also proposed at both extremities: click on it, do subsequent clicks and finish with a right-click; this allows to easily extend an existing line.

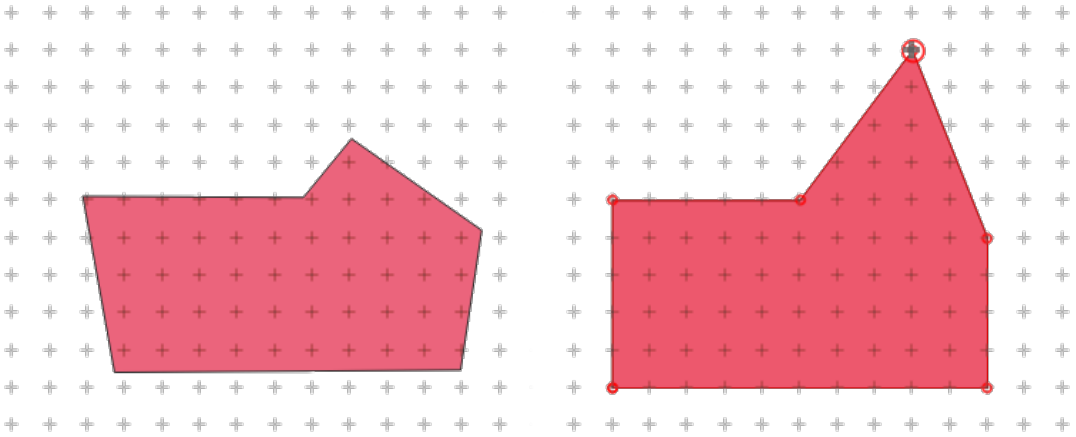


12.82 Fig. : Virtual nodes for adding vertices

- **Deleting vertices**: Select the vertices and click the `Delete` key. Deleting all the vertices of a feature generates, if compatible with the datasource, a geometryless feature. Note that this doesn't delete the complete feature, just the geometry part. To delete a complete feature use the Delete Selected tool.
- **Moving vertices**: Select all the vertices you want to move, click on a selected vertex or edge, and click on

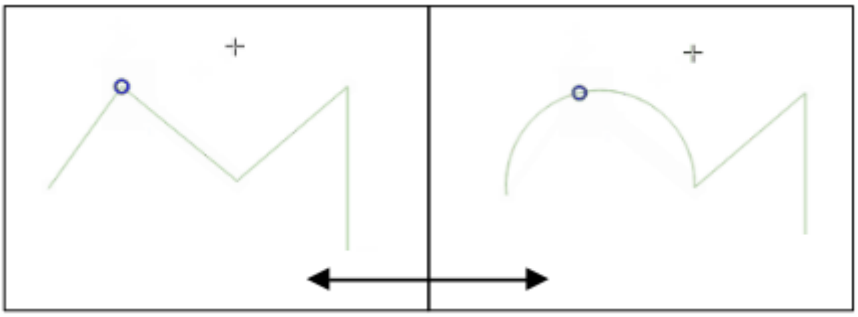
the desired new location. You can use the *snapping to feature capabilities* and the *Advanced Digitizing Panel* constraints for distance, angles, exact X and Y location before the second click. All the selected vertices will be translated.

However, if the *snap-to-grid* option is enabled, selected vertices are snapped to the closest grid intersection to their translated position. Unselected vertices are also moved to their closest grid intersection. There is no simple translation.



12.83 Fig. : Moving the top vertex snaps all the vertices to the grid

- **Converting adjacent segments to/from curve**: Select the center vertex of the segment you want to convert, hit the ○ letter key. If the vertex was in a curve, the curve is converted into straight lines. If the vertex was between two straight lines, they are converted into a curve. A first or a last vertex of a line can't be converted to a center vertex curve. The layer must be compatible with curve geometry type.



12.84 Fig. : Switch from curve to straight lines with ○ letter

Each change made with the vertex tool is stored as a separate entry in the *Undo* dialog. Remember that all operations support topological editing when this is turned on. On-the-fly projection is also supported.

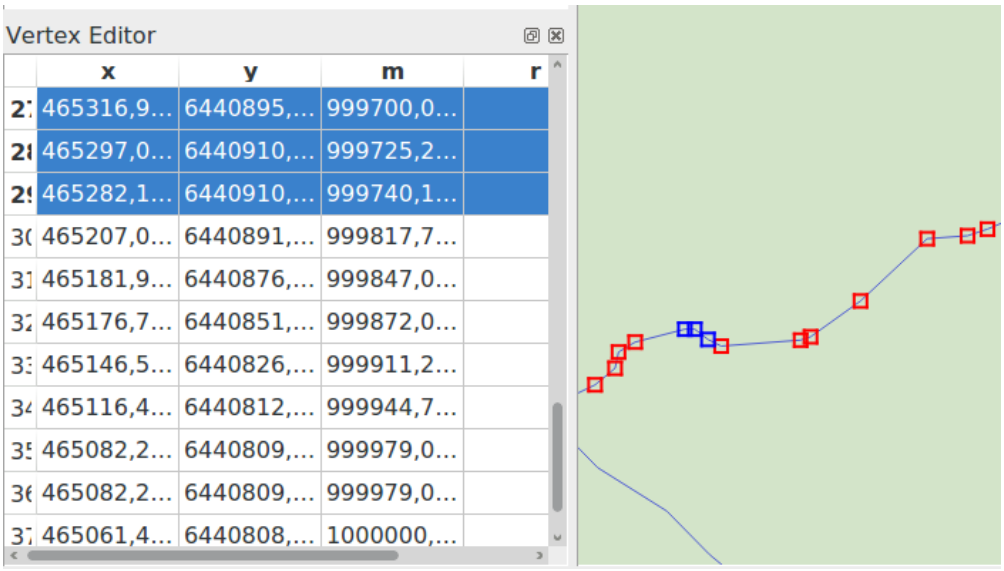
The Vertex Editor Panel

With enabling a vertex tool, you also open the *Vertex Editor* panel. Right-clicking over a feature fills the panel with the list of all the vertices of the feature with their *x*, *y* (*z*, *m* if applicable) coordinates and *r* (for the radius, in case of circular geometry). The feature is also made exclusive for editing, meaning that the edit of any other features is disabled:

- Selecting a row in the table does select the corresponding vertex in the map canvas, and vice versa.
- Clicking or dragging over the map canvas will only select or move vertices and segments of that feature

- Change a coordinate in the table and the vertex position is updated. This is a convenient way to edit Z coordinate or M value on vertices.
- You can also select multiple rows and delete them altogether.
- New vertices can only be added to the bound feature

If you do not want the *Vertex Editor* panel to immediately show each time you interact with vertex tools (and potentially hide other panels or disturb panels placement), uncheck the *Auto-open table* entry in the ☰ Options menu at the top of the panel. You can then also close the panel. To reopen the panel, you would need to right-click over a panel or toolbar and select it in the list or tick the *Show vertex editor* entry in the *Digitizing toolbar*.



12.85 Fig. : Vertex editor panel showing selected nodes

Rules of Z coordinate or M value assignment

Digitizing 3D vector features or features with M value is not that different from (X,Y) 2D layers'. Tools and options described in this chapter are still available and help you place the vertex or point in a planar environment. Then you may need to handle the Z coordinate (or M value) assignment:

- By default, QGIS will assign to new vertices the *Default Z value* (respectively *Default M value*) set in the *Settings* ► *Options* ► *Digitizing* tab. If the *Advanced Digitizing Panel* is in use, then the value is taken from its *z* (respectively *m*) widget.
- When snapping to a vertex, the new or moved vertex takes the snapped one's Z or M value.
- When snapping to a segment while the topological editing is on, then the new vertex Z or M value is interpolated along the segment.
- If the *z* (respectively *m*) widget of the *Advanced Digitizing Panel* is locked, then its value is applied to the vertex, taking precedence over any snapped vertex or segment Z or M value.

To edit Z or M values of an existing feature, you can use the *Vertex editor panel*. To create features with custom Z or M values you may want to rely on the *Advanced Digitizing Panel*.

Cutting, Copying and Pasting Features

Selected features can be cut, copied and pasted between layers in the same QGIS project, as long as destination layers are set to Toggle editing beforehand.

Patarimas

Transform polygon into line and vice-versa using copy/paste

Copy a line feature and paste it in a polygon layer: QGIS pastes in the target layer a polygon whose boundary corresponds to the closed geometry of the line feature. This is a quick way to generate different geometries of the same data.

Features can also be pasted to external applications as text. That is, the features are represented in CSV format, with the geometry data appearing in the OGC Well-Known Text (WKT) format. WKT and GeoJSON features from outside QGIS can also be pasted to a layer within QGIS.

When would the copy and paste function come in handy? Well, it turns out that you can edit more than one layer at a time and copy/paste features between layers. Why would we want to do this? Say we need to do some work on a new layer but only need one or two lakes, not the 5,000 on our `big_lakes` layer. We can create a new layer and use copy/paste to plop the needed lakes into it.

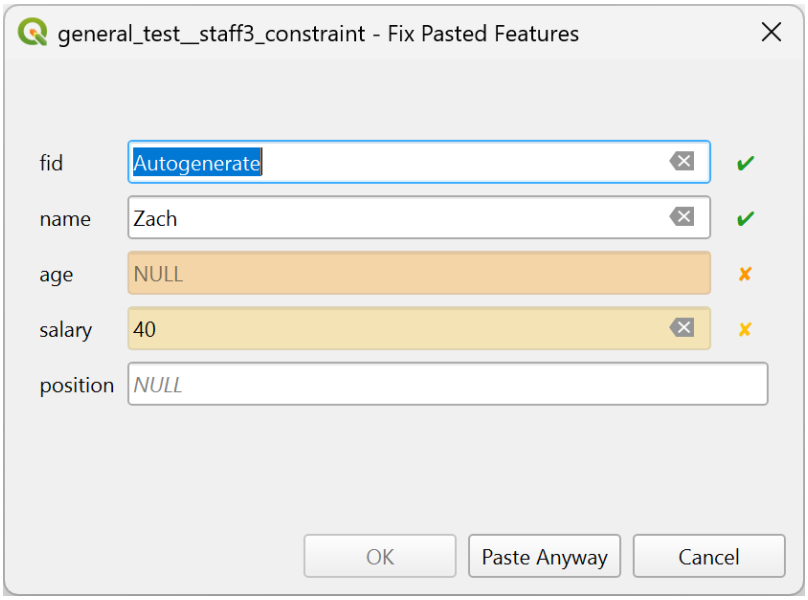
As an example, we will copy some lakes to a new layer:

1. Load the layer you want to copy from (source layer)
2. Load or create the layer you want to copy to (target layer)
3. Start editing for target layer
4. Make the source layer active by clicking on it in the legend
5. Use the Select Features by area or single click tool to select the feature(s) on the source layer
6. Click on the Copy Features tool
7. Make the destination layer active by clicking on it in the legend
8. Click on the Paste Features tool
9. Stop editing and save the changes

What happens if the source and target layers have different schemas (field names and types are not the same)? QGIS populates what matches and ignores the rest. If you don't care about the attributes being copied to the target layer, it doesn't matter how you design the fields and data types. If you want to make sure everything - the feature and its attributes - gets copied, make sure the schemas match.

When you paste features into another layer, QGIS checks the attribute values against the *constraints* defined in the destination layer (for example, `NOT NULL`, `Unique`, or expression constraints). If one or more pasted features contain invalid attribute values that do not meet these constraints, a dialog will appear, listing all affected features and fields. When pasting a single feature, you can decide whether to:

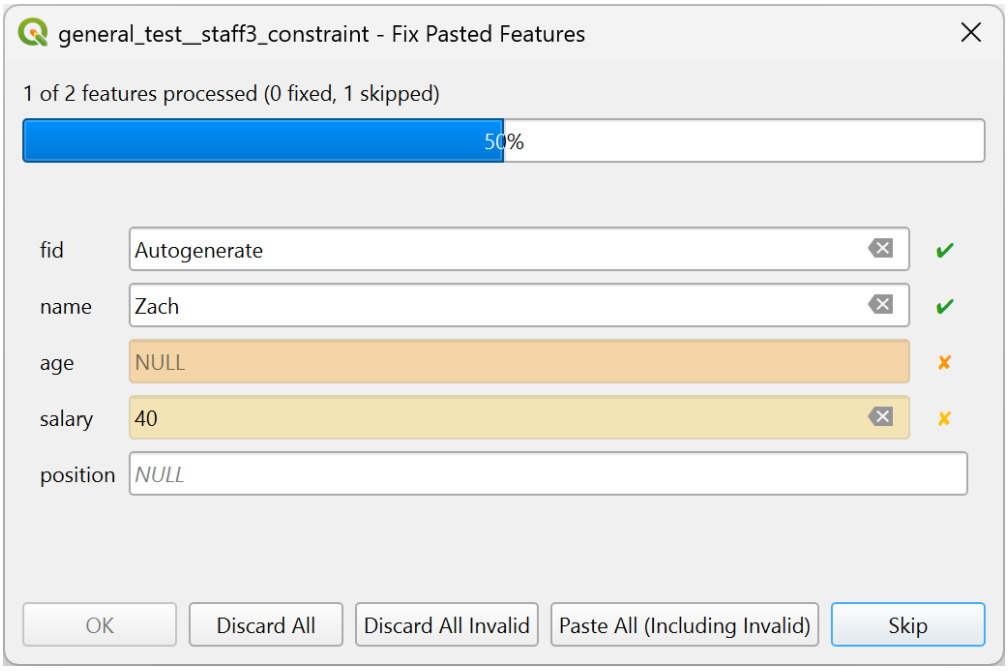
- Edit the invalid values directly in the dialog. The corrected values will be used when pasting.
- *Paste Anyway*: the feature is pasted as-is, even with invalid values.



12.86 Fig. : Dialog for handling invalid attribute values when pasting a single feature

When pasting multiple features, the dialog lists all invalid fields across all features, allowing bulk review and correction. You can decide per field or per feature whether to:

- Edit the invalid values directly in the dialog. The corrected values will be used when pasting.
- *Discard All*: no features pasted.
- *Discard All Invalid*: pasted only features with valid attribute values.
- *Paste All (Including Invalid)*: all features are pasted as-is, even with invalid values.
- *Skip*: skip the specific feature, allowing proceeding with the rest.



12.87 Fig. : Dialog for handling invalid attribute values when pasting multiple features

Pastaba

Congruency of Pasted Features

If your source and destination layers use the same projection, then the pasted features will have geometry identical to the source layer. However, if the destination layer is a different projection, then QGIS cannot guarantee the geometry is identical. This is simply because there are small rounding-off errors involved when converting between projections.

Patarimas

Copy string attribute into another

If you have created a new column in your attribute table with type ‚string' and want to paste values from another attribute column that has a greater length the length of the column size will be extended to the same amount. This is because the GDAL Shapefile driver knows to auto-extend string and integer fields to dynamically accommodate for the length of the data to be inserted.

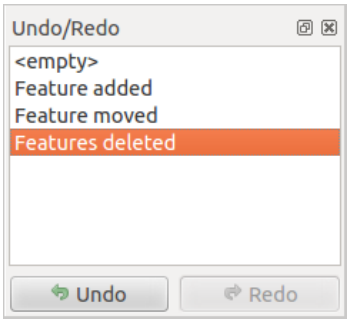
Deleting Selected Features

If we want to delete an entire feature (attribute and geometry), we can do that by first selecting the geometry using the regular Select Features by area or single click tool. Selection can also be done from the attribute table. Once you have the selection set, press `Delete` or `Backspace` key or use the Delete Selected tool to delete the features. Multiple selected features can be deleted at once.

The Cut Features tool on the digitizing toolbar can also be used to delete features. This effectively deletes the feature but also places it on a „spatial clipboard". So, we cut the feature to delete. We could then use the Paste Features tool to put it back, giving us a one-level undo capability. Cut, copy, and paste work on the currently selected features, meaning we can operate on more than one at a time.

Undo and Redo

The Undo and Redo tools allows you to undo or redo vector editing operations. There is also a dockable widget, which shows all operations in the undo/redo history (see Fig. 12.88). This widget is not displayed by default; it can be displayed by right-clicking on the toolbar and activating the *Undo/Redo Panel* checkbox. The Undo/Redo capability is however active, even if the widget is not displayed.



12.88 Fig. : Redo and Undo digitizing steps

When Undo is hit or `Ctrl+Z` (or `Cmd+Z`) pressed, the state of all features and attributes are reverted to the state before the reverted operation happened. Changes other than normal vector editing operations (for example, changes done by a plugin) may or may not be reverted, depending on how the changes were performed.

To use the undo/redo history widget, simply click to select an operation in the history list. All features will be reverted to the state they were in after the selected operation.

Saving Edited Layers

When a layer is in editing mode, any changes remain in the memory of QGIS. Therefore, they are not committed/saved immediately to the data source or disk. If you want to save edits to the current layer but want to continue editing without leaving the editing mode, you can click the Save Layer Edits button. When you turn editing mode off with Toggle editing (or quit QGIS for that matter), you are also asked if you want to save your changes or discard them.

If the changes cannot be saved (e.g., disk full, or the attributes have values that are out of range), the QGIS in-memory state is preserved. This allows you to adjust your edits and try again.

Patarimas

Data Integrity

It is always a good idea to back up your data source before you start editing. While the authors of QGIS have made every effort to preserve the integrity of your data, we offer no warranty in this regard.

Saving multiple layers at once

This feature allows the digitization of multiple layers. Choose *Save for Selected Layers* to save all changes you made in multiple layers. You also have the opportunity to *Rollback for Selected Layers*, so that the digitization may be withdrawn for all selected layers. If you want to stop editing the selected layers, *Cancel for Selected Layer(s)* is an easy way.

The same functions are available for editing all layers of the project.

Patarimas

Use transaction group to edit, save or rollback multiple layers changes at once

When working with layers from the same PostGreSQL database, activate the *Automatically create transaction groups where possible* option in *Project* ► *Properties…* ► *Data Sources* to sync their behavior (enter or exit the edit mode, save or rollback changes at the same time).

12.3.5 Advanced digitizing

12.4 Table : Vector layer advanced editing toolbar

Piktograma	Tikslas	Piktograma	Tikslas
	Enable Advanced Digitizing Tools		
	Move Feature(s)		Copy and Move Feature(s)
	Rotate Feature(s)		Simplify Feature
	Scale Feature		
	Add Ring		Add Part
	Fill Ring		Swap direction
	Delete Ring		Delete Part
	Offset Curve		Reshape Features
	Split Parts		Split Features
	Merge Attributes of Selected Features		Merge Selected Features
	Rotate Point Symbols		Offset Point Symbols
	Trim or Extend Feature		

Move Feature(s)

The Move Feature(s) tool allows you to move existing features:

1. Select the feature(s) to move.
2. Click on the map canvas to indicate the origin point of the displacement; you can rely on snapping capabilities to select an accurate point.

   You can also take advantages of the *advanced digitizing constraints* to accurately set the origin point coordinates. In that case:

   1. First click on the button to enable the panel.
   2. Type `x` and enter the corresponding value for the origin point you'd like to use. Then press the button next to the option to lock the value.
   3. Do the same for the `y` coordinate.
   4. Click on the map canvas and your origin point is placed at the indicated coordinates.
3. Move over the map canvas to indicate the destination point of the displacement, still using snapping mode or, as above, use the advanced digitizing panel which would provide complementary `distance` and `angle` placement constraints to place the end point of the translation.
4. Click on the map canvas: the whole features are moved to new location.

Likewise, you can create a translated copy of the feature(s) using the Copy and Move Feature(s) tool.

Pastaba

If no feature is selected when you first click on the map canvas with any of the *Move Feature(s)* or *Copy and Move Feature(s)* tools, then only the feature under the mouse is affected by the action. So, if you want to move several features, they should be selected first.

Rotate Feature(s)

Use the Rotate Feature(s) tool to rotate one or multiple features in the map canvas:

1. Press the Rotate Feature(s) icon
2. Then click on the feature to rotate. The feature's centroid is referenced as rotation center, a preview of the rotated feature is displayed and a widget opens showing the current *Rotation* angle.
3. Click on the map canvas when you are satisfied with the new placement or manually enter the rotation angle in the text box. You can also use the *Snap to* ° box to constrain the rotation values.
4. If you want to rotate several features at once, they shall be selected first, and the rotation is by default around the centroid of their combined geometries.

You can also use an anchor point different from the default feature centroid: press the `Ctrl` button, click on the map canvas and that point will be used as the new rotation center.

If you hold `Shift` before clicking on the map, the rotation will be done in 45 degree steps, which can be modified afterwards in the user input widget.

To abort feature rotation, press the `ESC` button or click on the Rotate Feature(s) icon.

Scale Feature

The Scale Feature tool is similar to the Rotate feature. Though instead of performing a rotation of selected features, it rescales their geometry. The change is performed in relation to the anchor point and the scale ratio can be manually specified in the widget that appears in the upper corner of the canvas.

Simplify Feature

The Simplify Feature tool allows you to interactively reshape a line or polygon geometry by reducing or densifying the number of vertices, as long as the geometry remains valid:

1. Select the Simplify Feature tool.
2. Click on the feature or drag a rectangle over the features.
3. A dialog pops up allowing you to define the *Method* to apply, ie whether you would like to:
   - *simplify the geometry*, meaning less vertices than the original. Available methods are `Simplify by distance`, `Simplify by snapping to grid` or `simplify by area (Visvalingam)`. You'd then need to indicate the value of *Tolerance* in `Layer units`, `Pixels` or `map units` to use for simplification. The higher the tolerance is the more vertices can be deleted.
   - or *densify the geometries* with new vertices thanks to the `Smooth` option: for each existing vertex, two vertices are placed on each of the segments originated from it, at an *Offset* distance representing the percentage of the segment length. You can also set the number of *Iterations* the placement would be processed: the more iterations, the more vertices and smoother is the feature.

   Settings that you used will be saved when leaving a project or an edit session. So you can go back to the same parameters the next time you simplify a feature.

4. A summary of the modifications that would apply is shown at the bottom of the dialog, listing number of features and number of vertices (before and after the operation and the ratio the change represents). Also, in the map canvas, the expected geometry is displayed over the existing one, using the rubberband color.
5. When the expected geometry fits your needs, click *OK* to apply the modification. Otherwise, to abort the operation, you can either press *Cancel* or right-click in the map canvas.

Pastaba

Unlike the feature simplification option in *Settings* ► *Options* ► *Rendering* menu which simplifies the geometry just for rendering, the Simplify Feature tool permanently modifies feature's geometry in data source.

Add Part

You can Add Part to a selected feature generating a multipoint, multiline or multipolygon feature. The new part must be digitized outside the existing one which should be selected beforehand.

The Add Part can also be used to add a geometry to a geometryless feature. First, select the feature in the attribute table and digitize the new geometry with the Add Part tool.

Pastaba

Order of vertices in polygon parts

Unlike the OGC standards, QGIS doesn't constrain vertices of the exterior boundary of a polygon feature to be ordered counterclockwise. Thus, you can find both directions in a layer. However, every parts of the same multipolygon feature will have their outer vertices ordered following the same direction.

You can however use the *Force right-hand-rule* algorithm to constrain features of a layer to have vertices of their outer boundaries ordered in the clockwise direction.

Delete Part

The Delete Part tool allows you to delete parts from multifeatures (e.g., to delete polygons from a multi-polygon feature). This tool works with all multi-part geometries: point, line and polygon. Furthermore, it can be used to totally remove the geometric component of a feature. To delete a part, simply click within the target part.

Add Ring

You can create ring polygons using the Add Ring icon in the toolbar. This means that inside an existing area, it is possible to digitize further polygons that will occur as a ‚hole', so only the area between the boundaries of the outer and inner polygons remains as a ring polygon.

To add a ring:

1. Select the feature(s) to modify
2. Activate the Add Ring tool
3. Draw a polygon within the selected geometries, using the aforementioned *techniques*. A hole appears in the selected geometries.

4. If no geometry is selected when the ring is drawn, then a hole is added to each of the polygons the ring is drawn over.

> **Pastaba**
>
> **Order of vertices in polygon rings**
>
> Unlike the OGC standards, QGIS doesn't constrain vertices of the exterior boundary of a polygon feature to be ordered counterclockwise. Thus, you can find both directions in a layer. However, every rings of the same (multi)polygon feature will have their vertices ordered in the opposite direction to the outer boundary's.
>
> You can however use the *Force right-hand-rule* algorithm to constrain features of a layer to have vertices of their outer boundaries ordered in the clockwise direction, and vertices of their interior rings ordered in the counter-clockwise direction.

Fill Ring

The Fill Ring tool helps you create polygon feature that totally falls within another one without any overlapping area; that is the new feature covers a hole within the existing one. To create such a feature:

1. Select the Fill Ring tool.
2. Draw a new polygon over the existing feature: QGIS adds a ring to its geometry (like if you used the Add Ring tool) and creates a new feature whose geometry matches the ring (like if you *traced* over the interior boundaries with the Add polygon feature tool).
3. Or alternatively, if the ring already exists on the feature, place the mouse over the ring and left-click while pressing `Shift`: a new feature filling the hole is drawn at that place.

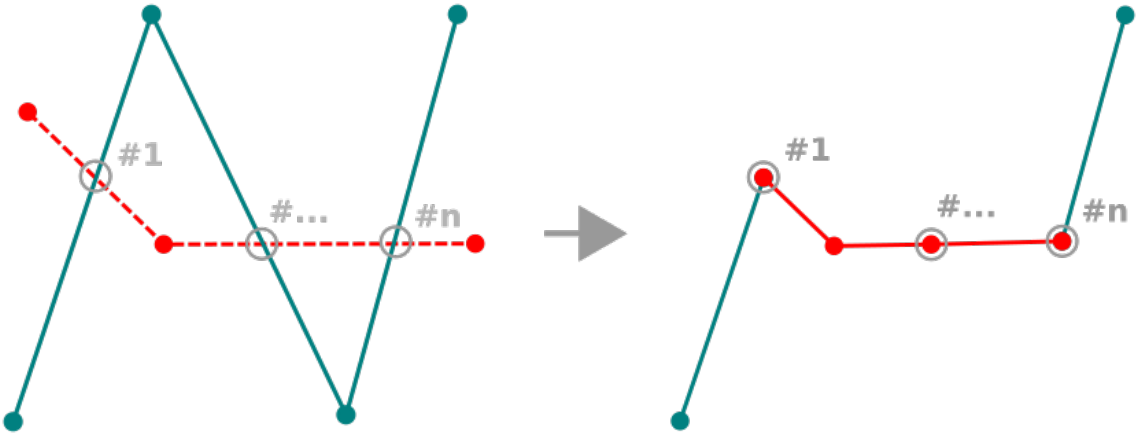
   The *Feature Attributes* form of the new feature opens, pre-filled with values of the „parent" feature and/or *fields constraints*.

Delete Ring

The Delete Ring tool allows you to delete rings within an existing polygon, by clicking inside the hole. This tool only works with polygon and multi-polygon features. It doesn't change anything when it is used on the outer ring of the polygon.

Reshape Features

You can reshape line and polygon features using the Reshape Features tool on the toolbar. For lines, it replaces the line part from the first to the last intersection with the original line.



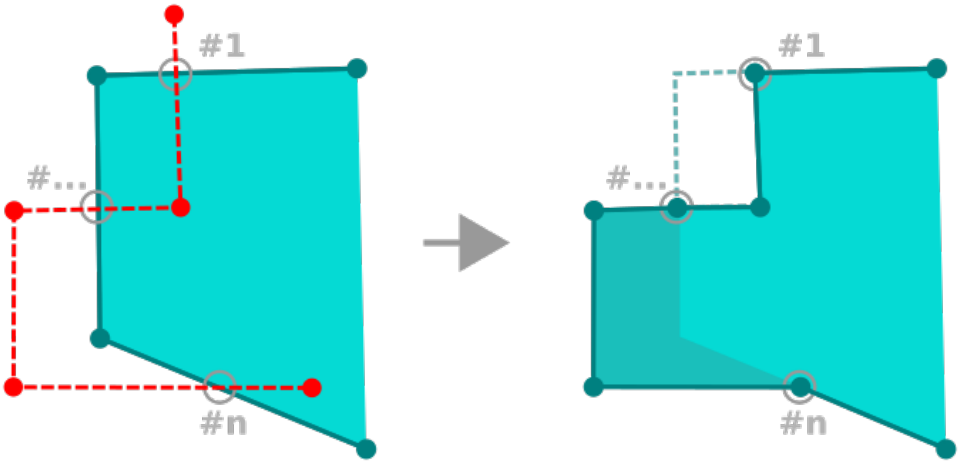
12.89 Fig. : Reshape line

Patarimas

Extend linestring geometries with reshape tool

Use the Reshape Features tool to extend existing linestring geometries: snap to the first or last vertex of the line and draw a new one. Validate and the feature's geometry becomes the combination of the two lines.

For polygons, it will reshape the polygon's boundary. For it to work, the reshape tool's line must cross the polygon's boundary at least twice. To draw the line, click on the map canvas to add vertexes. To finish it, just right-click. Like with the lines, only the segment between the first and the last intersections is considered. The reshape line's segments that are inside the polygon will result in cropping it, where the ones outside the polygon will extend it.



12.90 Fig. : Reshape polygon

With polygons, reshaping can sometimes lead to unintended results. It is mainly useful to replace smaller parts of a polygon, not for major overhauls, and the reshape line is not allowed to cross several polygon rings, as this would generate an invalid polygon.

Pastaba

The reshape tool may alter the starting position of a polygon ring or a closed line. So, the point that is represented ‚twice' will not be the same any more. This may not be a problem for most applications, but it is something to consider.

Offset Curves

The Offset Curve tool creates parallel shifts of line layers. The tool can be applied to the edited layer (the geometries are modified) or also to background layers (in which case it creates copies of the lines / rings and adds them to the edited layer). It is thus ideally suited for the creation of distance line layers. The *User Input* dialog pops-up, showing the displacement distance and other settings.

To create a shift of a line or polygon layer, you must first go into editing mode and activate the Offset Curve tool. Then click on a feature to shift it. Move the mouse and click where wanted or enter the desired distance in the user input widget. Holding `Ctrl` during the 2nd click will make an offset copy. Your changes may then be saved with the Save Layer Edits tool.

For geometries on background layers make sure that snapping is on and hold `Ctrl` to select the geometry from the background. Also hold `Ctrl` when doing the second click. Geometries will be converted to the target layer geometry type.

QGIS options dialog (Digitizing tab then **Curve offset tools** section) or the icon in the *User Input* dialog allows you to configure *some parameters* like **Join style**, **Quadrant segments**, **Miter limit** and **End cap style**.

Reverse Line

Changing the direction of a line geometry can be useful for cartographical purposes (e.g., for orienting symbols or labels that follow the line direction) or when preparing for network analysis where the direction of flow or travel matters.

This tool works only on line layers. For multipart lines, you can reverse individual parts without affecting the other parts of the feature.

To change a line direction:

1. Select a line layer in the *Layers* panel and make it editable.
2. Activate the reverse line tool by clicking Reverse line in the *Advanced Digitizing* toolbar.
3. Move the pointer over a line feature. The part under the cursor is highlighted.
4. Click on the line: the direction of the line (or part) is reversed.
5. You can reverse other lines by clicking on them.

Patarimas

For bulk line direction changes, use the *Reverse line direction* processing algorithm which can reverse all lines in a layer or a selection at once.

Split Features

Use the Split Features tool to split a feature into two or more new and independent features, ie. each geometry corresponding to a new row in the attribute table.

To split line or polygon features:

1. Select the Split Features tool.
2. Draw a line across the feature(s) you want to split. If a selection is active, only selected features are split. The original feature is then assigned the biggest geometry resulting from the splitting, and new features are created for the remaining parts. Fields of the features are filled/updated according to the datasource provider rules or their *splitting policy*.
3. You can then as usual modify any of the attributes of any resulting feature.

Patarimas

Split a polyline into new features in one-click

Using the Split Features tool, snap and click on an existing vertex of a polyline feature to split that feature into two new features.

Split parts

In QGIS it is possible to split the parts of a multi part feature so that the number of parts is increased. Just draw a line across the part you want to split using the Split Parts icon.

Patarimas

Split a polyline into new parts in one-click

Using the Split Parts tool, snap and click on an existing vertex of a polyline feature to split the feature into two new polylines belonging to the same feature.

Merge selected features

The Merge Selected Features tool allows you to create a new feature by merging existing ones: their geometries are merged to generate a new one. If features don't have common boundaries, a multipolygon/multipolyline/multipoint feature is created.

1. First, select the features you'd like to combine.
2. Then press the Merge Selected Features button.
3. In the new dialog, the *Merge* line at the bottom of the table shows the attributes of the resulting feature. If any *Merge policy* have been defined, they are automatically applied to populate this row. However, you can override these values manually. Modify the values in the following ways:
   - manually replacing the value in the corresponding cell;
   - selecting a row in the table and pressing *Take attributes from selected feature* to use the values of this initial feature;
   - pressing the *Take attributes from the largest geometry* to use the attributes from the longest line feature, the largest polygon, or the multipoints with the most parts;

- pressing *Skip all fields* to use empty attributes;
- expanding the drop down menu at the top of the table, select any of the above options to apply to the corresponding field only. There, you can also choose to aggregate the initial features attributes (Minimum, Maximum, Median, Sum, Count, Concatenation… depending on the type of the field. see *Statistinės suvestinės skydelis* for the full list of functions).

> **Pastaba**
>
> If the layer has default values or clauses present on fields, these are used as the initial value for the merged feature.

4. Press *OK* to apply the modifications. A single (multi)feature is created in the layer, replacing the previously selected ones.

Merge attributes of selected features

The Merge Attributes of Selected Features tool allows you to apply same attributes to features without merging their boundaries. The dialog is the same as the `Merge Selected Features` tool's except that unlike that tool, selected objects are kept with their geometry while some of their attributes are made identical.

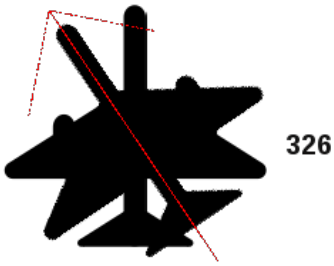
Rotate Point Symbols

The Rotate Point Symbols allows you to individually change the rotation of point symbols in the map canvas.

1. First, you need to indicate the field to store the rotation value in. This is made by assigning a field to the symbol *data-defined* rotation property:
   1. In the *Layer Properties* ► *Symbology* dialog, browse to the symbol editor dialog.
   2. Click the *Data-defined override* widget near the *Rotation* option of the top *Marker* level (preferably) of the symbol layers.
   3. Choose a field in the *Field Type* combobox. Values of this field are hence used to rotate each feature's symbol accordingly.

      You can also check the Store data in project entry to generate an *auxiliary data storage* field to control the rotation value.

> **Pastaba**
>
> **Make sure that the same field is assigned to all the symbol layers**
>
> Setting the data-defined rotation field at the topmost level of the symbol tree automatically propagates it to all the symbol layers, a prerequisite to perform graphical symbol rotation with the *Rotate Point Symbols* tool. Indeed, if a symbol layer does not have the same field attached to its rotation property, the tool will not work.



12.91 Fig. : Rotating a point symbol

2. Then click on a point symbol in the map canvas with the Rotate Point Symbols tool
3. Move the mouse around. A red arrow with the rotation value will be visualized (see Fig. 12.91). If you hold the `Ctrl` key while moving, the rotation will be done in 15 degree steps.
4. When you get the expected angle value, click again. The symbol is rendered with this new rotation and the associated field is updated accordingly.

   You can right-click to abort symbol rotation.

Offset Point Symbols

The Offset Point Symbols allows you to interactively change the rendered position of point symbols in the map canvas. This tool behaves like the Rotate Point Symbols tool except that it requires you to connect a field to the data-defined *Offset (X,Y)* property of each layer of the symbol. The field will then be populated with the offset coordinates for the features whose symbol is moved in the map canvas.

1. Associate a field to the data-defined widget of the *Offset (X,Y)* property of the symbol. If the symbol is made with many layers, you may want to assign the field to each of them
2. Select the Offset Point Symbols tool
3. Click a point symbol
4. Move to a new location
5. Click again. The symbol is moved to the new place. Offset values from the original position are stored in the linked field.

   You can right-click to abort symbol offset.

Pastaba

The Offset Point Symbols tool doesn't move the point feature itself; you should use the Vertex Tool (Current Layer) or Move Feature tool for this purpose.

Trim/Extend Feature

The Trim/Extend tool allows you to shorten or lengthen segments of a (multi)line or (multi)polygon geometry to converge with a selected segment (the cutting line). This results in a modified geometry with a vertex snapped to a reference segment or in its prolongation. Depending on how the selected geometries are placed in relation to each other, the tool will either:

- **Trim**: removes parts of the line segment or polygon boundary, beyond the cutting line
- **Extend**: extends polygon boundaries or line segments so that they can snap to the cutting line.

In order to trim or extend existing geometries:

1. Select the Trim/Extend tool
2. Click the reference segment, i.e., the segment with respect to which you want to extend or trim another segment. It doesn't need to be on the active layer. A red dotted line appears, visually extending the reference segment across the map canvas according to the active layer's CRS.
3. Hover over the target segment, i.e., the one you want to trim or extend. It does not need to be the last segment of the geometry, but has to be on the active layer. QGIS displays a preview of the feature's geometry with the segment limited to its intersection with the highlighted red dotted extending line.
    1. If an extension of the segment is required, click anywhere on the segment.
    2. In the case of a trim, click the part that should remain.
    3. In either case, the feature's geometry is updated accordingly. When both segments are in 3D, the tool performs an interpolation on the limit segment to get the Z value.
4. If you need to use the same reference segment for trimming or extending many features:
    1. Press `Shift` while selecting the reference segment.
    2. Click consecutively on the segments to modify and each will be trimmed or extended accordingly.

Pastaba

Snapping is automatically enabled when this tool is activated. Your original snapping settings will be restored once the tool is deactivated.

Dėmesio

Pay attention to the modified geometry while using the Trim/Extend tool. Depending on the inputs, it can create invalid geometries, potentially resulting in failure at layer saving.

12.3.6 Shape digitizing

The *Shape Digitizing* toolbar offers a set of tools to draw lines or polygons features of regular shape. It is synchronized with the Digitize Shape *geometry drawing method* you can select on the *Digitizing Toolbar*. To use it:

1. Display the toolbar: *View* ► *Toolbars* ► *Shape Digitizing*
2. Select a tool that creates or modifies the shape of a geometry, e.g. Add line feature, Add polygon feature, Add part, Add ring, Reshape Features, ...

3. The Digitize with segment button on the *Digitizing Toolbar* is enabled. The first time, you may need to switch it to the Digitize Shape in order to enable tools on the *Shape Digitizing* toolbar.
4. Pick a shape digitizing tool and draw.

Circular string by radius

The Circular string by radius button allows to add line or polygon features with a circular geometry, given two nodes on the curve and a radius:

1. Left click twice to place the two points on the geometry.
2. A *Radius* widget in the top right corner of the map canvas displays current radius (corresponding to distance between the points). Edit that field to the value you want.
3. An overview of the arcs matching these constraints is displayed while moving around the cursor. Right-click to validate when the expected arc is shown.
4. Add a new point to start shaping another arc.

Pastaba

Curved geometries are stored as such only in compatible data provider

Although QGIS allows to digitize curved geometries within any editable data format, you need to be using a data provider (e.g. PostgreSQL, Oracle Spatial, memory layer, GML or WFS) that supports curves to have features stored as curved, otherwise QGIS segmentizes the circular arcs.

Draw Circles

There is a set of tools for drawing circles. The tools are described below.

Circles are converted into circular strings. Therefore, as explained in *Circular string by radius*, if allowed by the data provider, it will be saved as a curved geometry, if not, QGIS will segmentize the circular arcs.

- Circle from 2 points: The two points define the diameter and the orientation of the circle. (Left-click, right-click)
- Circle from 3 points: Draws a circle from three known points on the circle. (Left-click, left-click, right-click)
- Circle by a center point and another point: Draws a circle with a given center and a point on the circle (Left-click, right-click). When used with the *The Advanced Digitizing panel* this tool can become a „Add circle from center and radius“ tool by setting and locking the distance value after first click.
- Circle from 3 tangents: Draws a circle that is tangential to three segments. **Note that you must activate snapping to segments** (See *Setting the snapping tolerance and search radius*). Click on a segment to add a tangent. If two tangents are parallel, the coordinates of the click on the first parallel tangent are used to determine the positioning of the circle. If three tangents are parallel, an error message appears and the input is cleared. (Left-click, left-click, right-click)
- Circle from 2 tangents and a point: Similar to circle from 3 tangents, except that you have to select two tangents, enter a radius and select the desired center.

Draw Ellipses

There is a set of tools for drawing ellipses. The tools are described below.

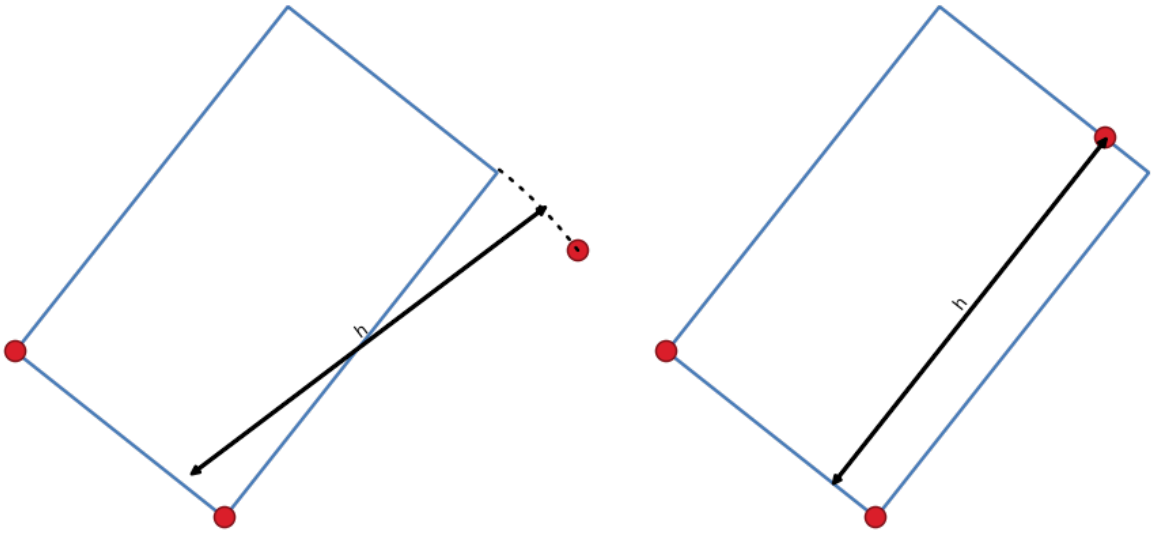
Ellipses cannot be converted as circular strings, so they will always be segmented.

- Ellipse from center and two points: Draws an ellipse with a given center, major axis and minor axis. (Left-click, left-click, right-click)
- Ellipse from center and a point: Draws an ellipse into a bounding box with the center and a corner. (Left-click, right-click)
- Ellipse from extent: Draws an ellipse into a bounding box with two opposite corners. (Left-click, right-click)
- Ellipse from foci: Draws an ellipse by 2 points for foci and a point on the ellipse. (Left-click, left-click, right-click)

Draw Rectangles

There is a set of tools for drawing rectangles. The tools are described below.

- Rectangle from center and a point: Draws a rectangle from the center and a corner. (Left-click, right-click)
- Rectangle from extent: Draws a rectangle from two opposite corners. (Left-click, right-click)
- Rectangle from 3 points (distance): Draws an oriented rectangle from three points. The first and second points determine the length and angle of the first edge. The third point determines the length of the other edge. One can use *The Advanced Digitizing panel* to set the length of the edges. (Left-click, left-click, right-click)
- Rectangle from 3 points (projected): Same as the preceding tool, but the length of the second edge is computed from the projection of the third point on the first edge. (Left-click, left-click, right-click)



12.92 Fig. : Draw rectangle from 3 points using distance (right) and projected (left)

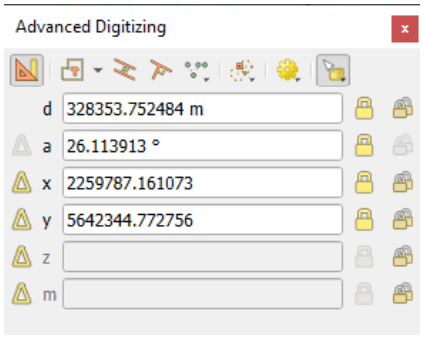
Draw Regular Polygons

There is a set of tools for drawing regular polygons. The tools are described below. Left-click to place the first point. A dialog appears, where you can set the number of polygon edges. Right-click to finish the regular polygon.

- Regular polygon from two points: Draws a regular polygon where the two points determine the length and angle of the first edge.
- Regular polygon from center and a point: Draws a regular polygon from the provided center point. The second point determines the angle and distance to the middle of an edge.
- Regular polygon from center and a corner: Same as the preceding tool, but the second point determines the angle and distante to a vertex.

12.3.7 The Advanced Digitizing panel

When capturing, reshaping, splitting new or existing geometries you also have the possibility to use the Advanced Digitizing panel. You can digitize lines exactly parallel or perpendicular to a particular angle or lock lines to specific angles. Furthermore, you can make a precise definition of your new geometry by entering X and Y coordinates as well as Z for 3D features, or M values.



12.93 Fig. : The Advanced Digitizing panel

The *Advanced Digitizing* panel can be opened either with a right-click on the toolbar, from *View* ► *Panels* ► menu or pressing `Ctrl+4`. Once the panel is visible, click the Enable advanced digitizing tools button to activate the set of tools.

Pastaba

The tools are not enabled if the map view is in geographic coordinates.

The aim of the Advanced Digitizing tool is to lock coordinates, lengths, and angles when moving the mouse during the digitalizing in the map canvas.

You can also create constraints with relative or absolute reference. Relative reference means that the next vertex constraints‘ values will be relative to the previous vertex or segment.

The toolbar

At the top of the *Digitizing panel*, you find the following buttons:

- Enable advanced digitizing tools
- Construction mode: allows to capture the clicks' positions to reuse as reference points to lock distance, angle, X, Y, Z or M relative values. More details at *Construction mode*.
- Parallel to draw a line parallel to an existing one (more at *Parallel and perpendicular lines*)
- Perpendicular to draw a line perpendicular to an existing one (more at *Parallel and perpendicular lines*)
- Construction Tools provides a couple of options that constrain the vertices placement based on extrapolated coordinates of existing elements:
  - *Line Extension*: hover over a segment and you get a purple dotted line extending the segment across the map canvas. You can snap the vertex anywhere on this virtual line.
  - *X/Y Point*: hover over a vertex and you get a purple dotted line along its X or Y coordinate, across the map canvas. You can snap the vertex anywhere on this virtual line. It is even possible to hover over two different vertices, generating virtual coordinate lines for both, and snap to their intersection.
- 2-circle Point Intersection: allows you to digitize a point or vertex at the intersection of two circles. (more at *2-circle point intersection*).
- Snap to common angles: when moving the cursor, displays a virtual line that you can snap to to add the next vertex. The snapping line is defined by the last added vertex and an (absolute or relative to previous segment) angle from a preset list (following steps of 0.1°, 0.5°, 1°, 5°, 10°, 15°, 18°, 22.5°, 30°, 45° or 90°). Choose *Do not snap to common angles* to disable this feature.

  Snapping to features can be used along with snapping to common angles for accurate digitizing. For a fine-grained control on how the target element to snap to is retained, you can indicate whether to prioritize snapping to features over common angles, and vice-versa under the *Snapping priority* entry. You can switch from one method to the other during the digitizing operation, and this avoids disabling any of the snapping options in the meantime. Press `N` (or `Shift+N`) during a digitizing operation to cycle through the angles list.
- Floater settings: if the *Show floater* item is checked, a contextual menu with digitizing information follows the cursor during digitizing. The values can be accessed using the *panel's shortcuts*, edited and Locked after validation (pressing `Enter`). The type of information to display can be selected in the bottom part of the menu:
  - *Show distance*
  - *Show angle*
  - *Show XY coordinates*
  - *Show Z value*
  - *Show M value*
  - *Show bearing/azimuth*
  - *Show common snapping angle*

Below the toolbar, you will find a number of text boxes whose value reflects by default the position or movement of the cursor in the map canvas. Editing these values helps you constrain the position of the items you edit:

- *d* for the distance from a reference position, usually the last edited vertex
- *a* for the angle (absolute or relative) from a reference position, usually the last edited segment
- *x* for the X coordinate of the pointer

- *y* for the Y coordinate of the pointer
- *z* for the default Z value or the Z coordinate of the vertex or segment under the pointer
- *m* for the default M value or the M value of the vertex or segment under the pointer

Klavišų kombinacijos

To speed up the use of Advanced Digitizing Panel, there are a couple of keyboard shortcuts available:

12.5 Table : Keyboard shortcuts of the Advanced Digitizing Panel tools

Key	Simple	`Ctrl+` or `Alt+`	`Shift+`
`D`	Set distance	Lock distance	
`A`	Set angle	Lock angle	Toggle relative angle to last segment
`X`	Set X coordinate	Lock X coordinate	Toggle relative X to last vertex
`Y`	Set Y coordinate	Lock Y coordinate	Toggle relative Y to last vertex
`Z`	Set Z coordinate	Lock Z coordinate	Toggle relative Z to last vertex
`M`	Set M value	Lock M value	Toggle relative M to last vertex
`C`	Toggle construction mode		
`P`	Toggle perpendicular and parallel modes		

Pastaba

Z coordinate and M value options are available only if compatible with the layer geometry dimension.

Absolute reference digitizing

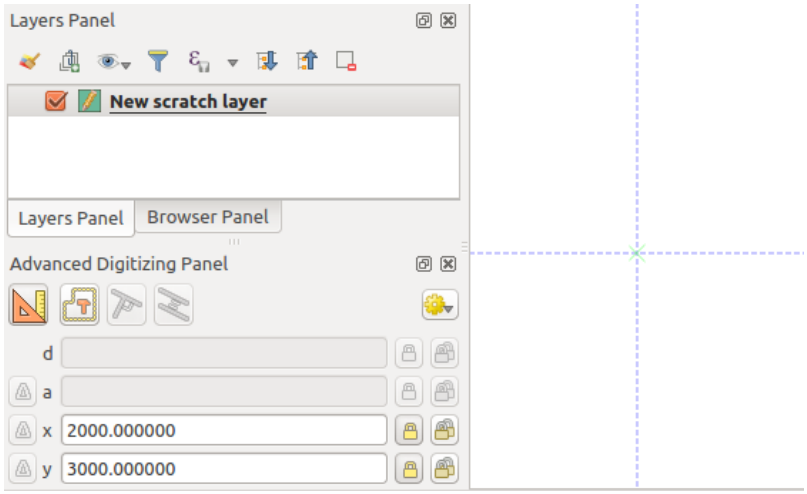
When drawing a new geometry from scratch, it is very useful to have the possibility to start digitizing vertexes at given coordinates.

For example, to add a new feature to a polygonal layer, click the button. You can enter the exact coordinates where you want to start editing the feature, i.e.:

1. Click the *x* text box (or use the `X` keyboard shortcut).
2. Type the X coordinate value you want and press `Enter` or click the button to their right to lock the mouse to the X axis on the map canvas.
3. Click the *y* text box (or use the `Y` keyboard shortcut).
4. Type the Y coordinate value you want and press `Enter` or click the button to their right to lock the mouse to the Y axis on the map canvas.
5. If the layer has Z coordinate or M values, the corresponding *z* or *m* widget is enabled and displays its default value, as set in *Settings* ► *Options* ► *Digitizing* tab.
    1. Click the *z* or *m* text box (or use respectively the `Z` or `M` keyboard shortcut).
    2. Type the coordinate value you want and press `Enter` or click the button to their right to lock the value in the widget.

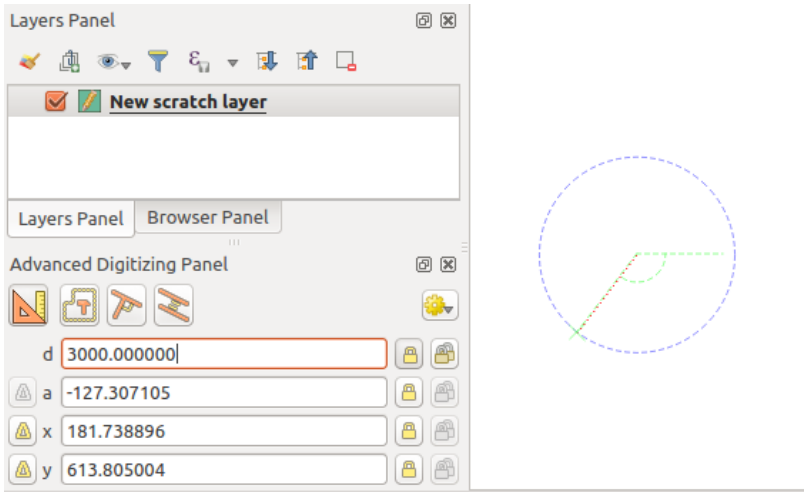
> **Pastaba**
>
> Read *Rules of Z coordinate or M value assignment* for details on how Z coordinate and M values are automatically determined from existing features.

6. Two blue dotted lines and a green cross identify the exact coordinates you entered. Click on the map canvas to add a vertex at the green cross position.



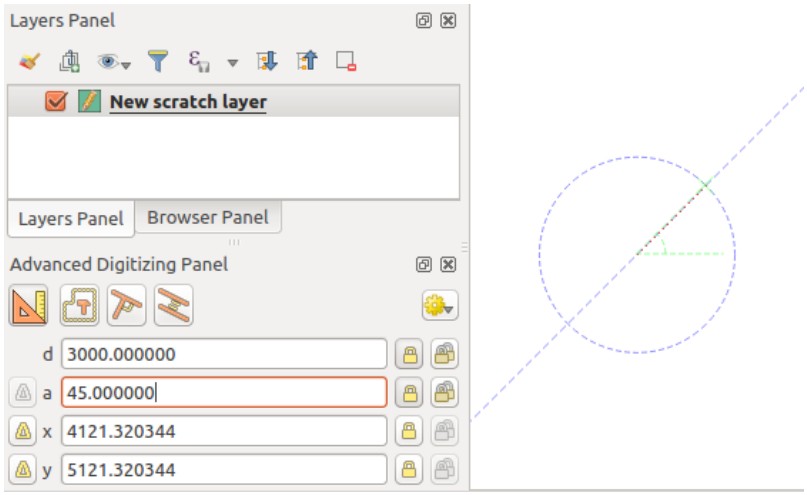
12.94 Fig. : Start drawing at given coordinates

7. You can proceed as above, adding a new set of coordinates for the next vertex, or switch to another *mode of digitizing* (e.g. segment, curve or stream).
8. If you want to draw a segment of a given length:
   1. Click the *d (distance)* text box (keyboard shortcut `D`)
   2. Type the distance value (in map units)
   3. Press `Enter` or click the button on the right to lock the mouse in the map canvas to the length of the segment. In the map canvas, the latest vertex is surrounded by a circle whose radius is the value entered in the distance text box. A cross on the circle shows the position of the next vertex if you click.



12.95 Fig. : Fixed length segment

9. You can also constrain the vertex position, setting the angle of the segment. As described before:
    1. Click the *a (angle)* text box (keyboard shortcut `A`)
    2. Type the angle value (in degrees)
    3. Press `Enter` or click the button on the right to lock it. A line going through the latest vertex and rotated based on the set angle appears in the map canvas and a cross on it shows the next vertex position if you click.



12.96 Fig. : Fixed angle segment

Patarimas

Pressing `Ctrl+<key>` or `Alt+<key>` automatically locks the target property and puts its value into edit. Modify, press `Enter` and you are done. Combined with the Toggle floater, this can be a real time saver, with keyboard digitizing.

Relative reference digitizing

Instead of using absolute values of angles or coordinates, you can also use values relative to the last digitized vertex or segment.

For angles, you can click the button on the left of the *a* text box (or press `Shift+A`) to toggle relative angles to the previous segment. With that option on, angles are measured between the last segment and the mouse pointer.

For coordinates, click the buttons to the left of the *x*, *y*, *z* or *m* text boxes (or press `Shift+<key>`) to toggle relative coordinates to the previous vertex. With these options on, coordinates measurement will consider the last vertex to be the origin of the set coordinates.

Continuous lock

Both in absolute or relative reference digitizing, angle, distance, X, Y, Z and M constraints can be locked continuously by clicking the *Continuous lock* buttons. Using continuous lock allows you to digitize several points or vertexes using the same constraints.

Construction mode

You can enable and disable *construction mode* by clicking on the Construction mode icon or with the `C` keyboard shortcut. While in construction mode, clicking the map canvas won't add new vertexes, but will capture the clicks' positions so that you can use them as reference points to then lock distance, angle or X, Y, Z, M relative values.

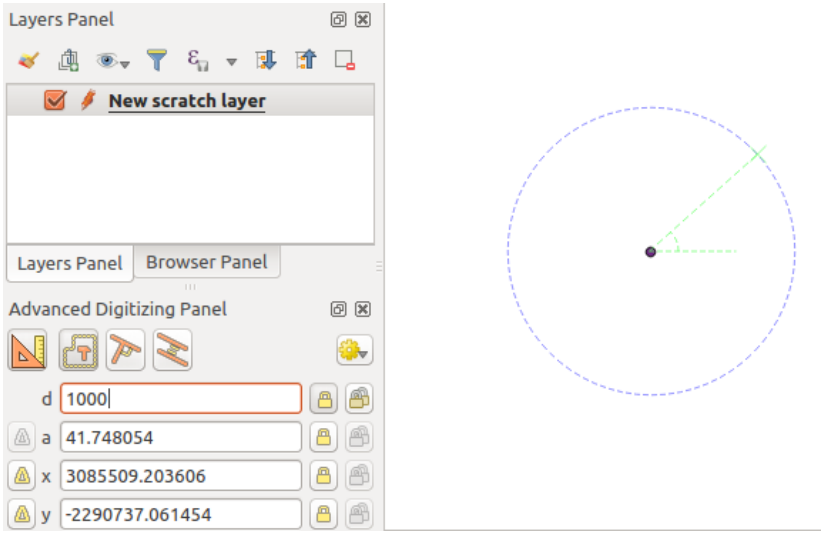
In the Construction mode you will find a drop-down menu where you can choose to:

- *Record Construction Guides*: All construction steps are rendered as dashed lines. Each step that you perform is visually represented, allowing you to trace the construction process. The guides are displayed as long as *Advanced Digitizing* remains active.
- *Show Construction Guides*: Allows you to make construction guides visible or hidden on the canvas. When enabled, all active guides are displayed, offering better spatial orientation and precision during the digitizing process.
- *Snap to Visible Construction Guides*: The guides are snap-able, allowing you to start new construction steps from any point along the existing guides.
- Choose *Clear Construction Guides* to remove all the guides from the canvas.

As an example, the construction mode can be used to draw some point at an exact distance from an existing point.

With an existing point in the map canvas and the snapping mode correctly activated, you can easily draw other points at given distances and angles from it. In addition to the button, you have to activate also the *construction mode* by clicking the Construction mode icon or with the `C` keyboard shortcut.

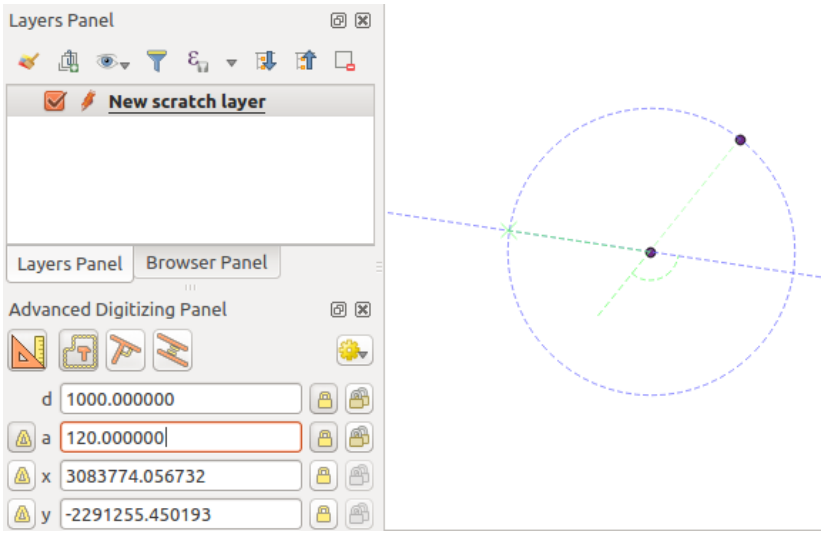
Click next to the point from which you want to calculate the distance and click on the *d* box (`D` shortcut) type the desired distance and press `Enter` to lock the mouse position in the map canvas:



12.97 Fig. : Distance from point

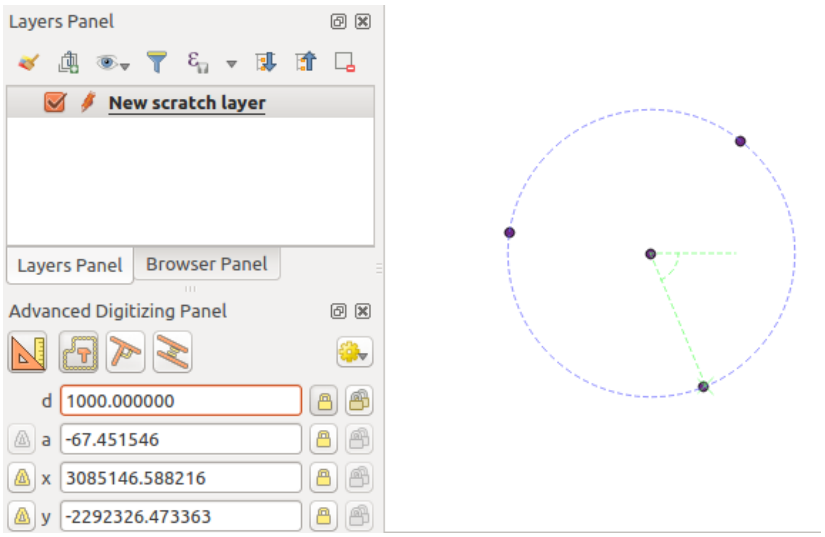
Before adding the new point, press `C` to exit the construction mode. Now, you can click on the map canvas, and the point will be placed at the distance entered.

You can also use the angle constraint to, for example, create another point at the same distance of the original one, but at a particular angle from the newly added point. Click the Construction mode icon or with the `C` keyboard shortcut to enter construction mode. Click the recently added point, and then the other one to set a direction segment. Then, click on the *d* text box (`D` shortcut) type the desired distance and press `Enter`. Click the *a* text box (`A` shortcut) type the angle you want and press `Enter`. The mouse position will be locked both in distance and angle.



12.98 Fig. : Distance and angle from points

Before adding the new point, press `C` to exit the construction mode. Now, you can click on the map canvas, and the point will be placed at the distance and angle entered. Repeating the process, several points can be added.



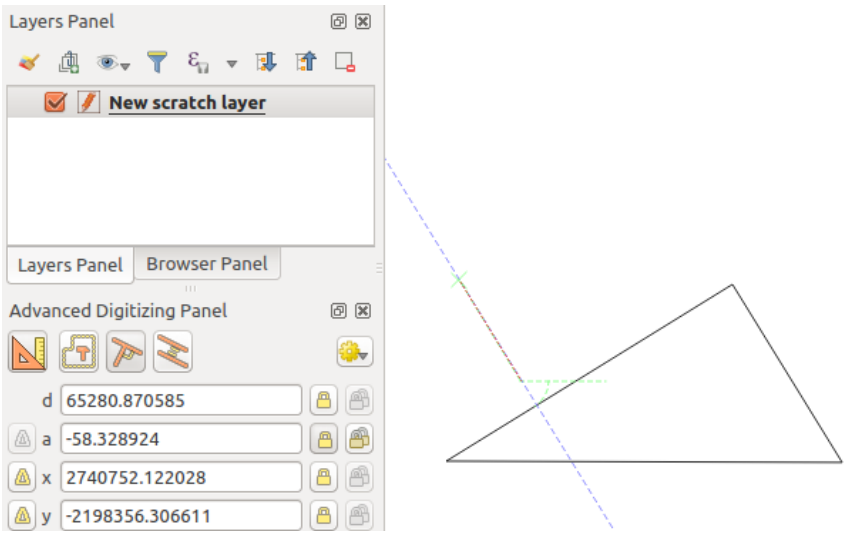
12.99 Fig. : Points at given distance and angle

Parallel and perpendicular lines

All the tools described above can be combined with the Perpendicular and Parallel tools. These two tools allow drawing segments perfectly perpendicular or parallel to another segment. The target segment can be on another layer, another feature within the layer or the feature being digitized (requires *self-snapping option*).

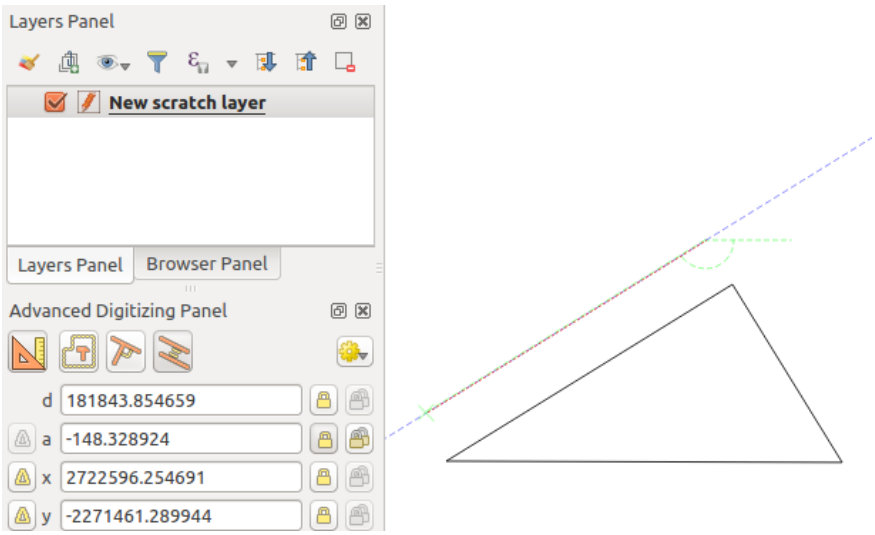
To draw a *perpendicular* segment:

1. First add one of the segment vertices.
2. Click the Perpendicular icon (keyboard shortcut `P`) to activate it.
3. Click on the segment that you want to be perpendicular to.
4. A virtual dotted line perpendicular to the segment through the previous vertex appears. The angle property is locked, constraining the next vertex on that line and, a cross indicates the projected position of the cursor on the line. Click to place the new vertex.



12.100 Fig. : Perpendicular digitizing

To draw a *parallel* segment, the steps are the same except that you need to click on the Parallel icon (keyboard shortcut `P` twice).



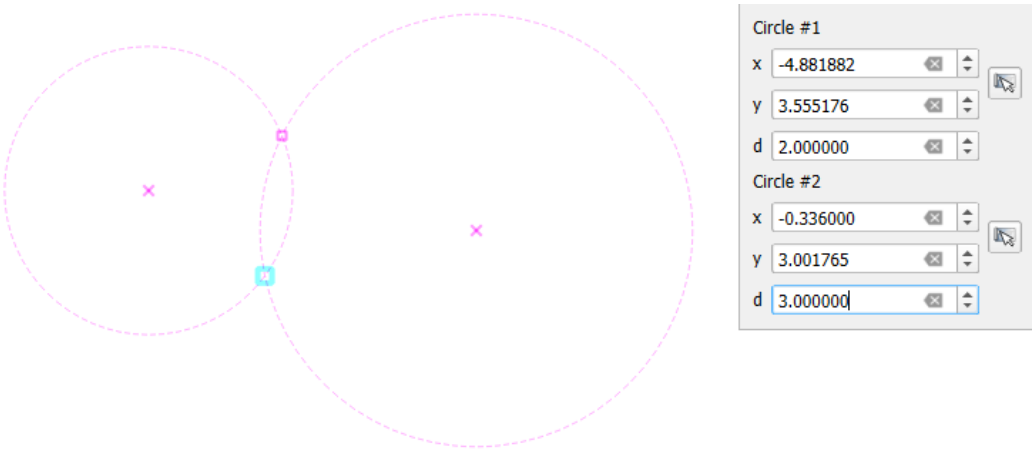
12.101 Fig. : Parallel digitizing

These two tools just find the right angle of the perpendicular and parallel angle and lock this parameter during your editing. Unlock the angle parameter to cancel their use in the middle of the process.

2-circle point intersection

To add a vertex at the intersection of two circles, follow these steps:

1. Click the 2-circle Point Intersection icon.
2. A dialog will open where you can define the parameters for *Circle #1* and *Circle #2*.
3. Click on the map canvas, and the tool will automatically calculate the *X* and *Y* coordinates for the centers of both circles.
4. Enter the distance *d* for each circle.
5. The tool will calculate and display the two intersection points of the circles.
6. Click one of the intersection points to add the new vertex.



12.102 Fig. : 2-circle Point Intersection

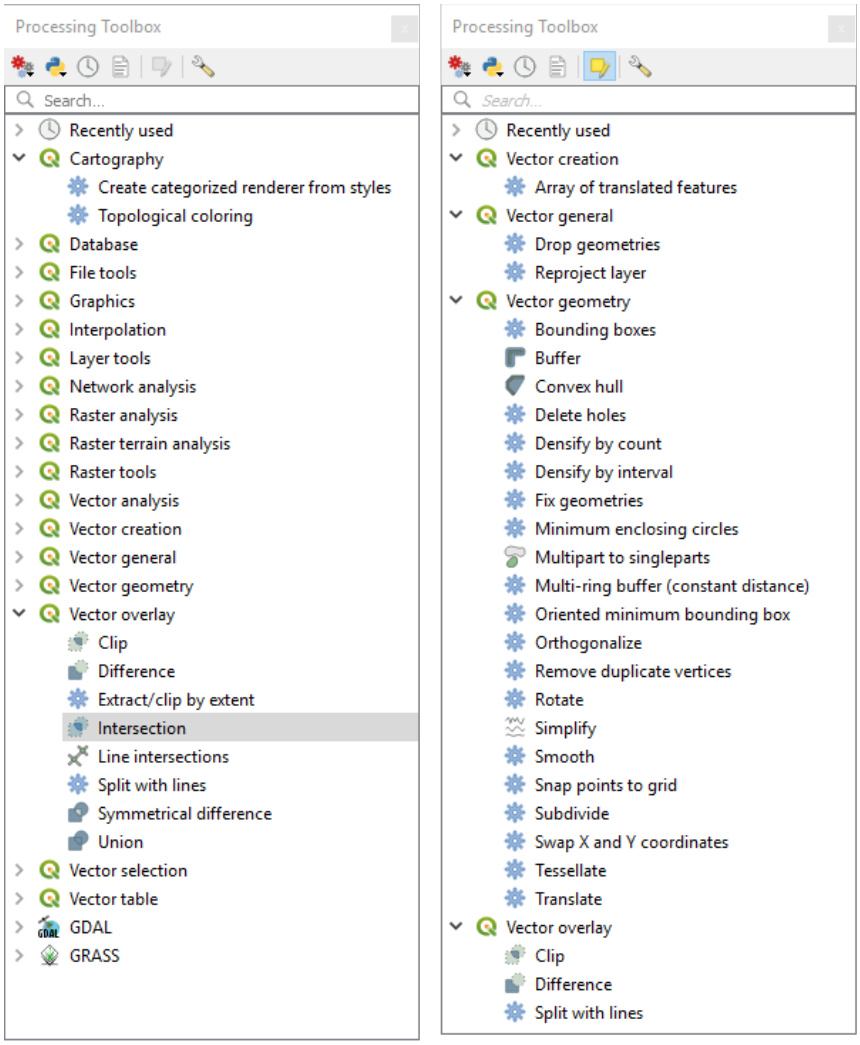
12.3.8 The Processing in-place layer modifier

The *Processing menu* provides access to a large set of tools to analyze and create new features based on the properties of the input features or their relations with other features (within the same layer or not). While the common behavior is to create new layers as outputs, some algorithms also allow modifications to the input layer. This is a handy way to automate multiple features modification using advanced and complex operations.

To edit features in-place:

1. Select the layer to edit in the *Layers* panel.
2. Select the concerned features. You can skip this step, in which case the modification will apply to the whole layer.
3. Press the Edit Features In-Place button at the top of the *Processing toolbox*. The list of algorithms is filtered, showing only those compatible with in-place modifications, i.e.:
   - They work at the feature source and not at the layer level.
   - They do not change the layer structure, e.g. adding or removing fields.

- They do not change the geometry type, e.g. from line to point layer.



12.103 Fig. : Processing algorithms: all (left) vs polygon in-place editors (right)

4. Find the algorithm you'd like to run and double-click it.

> **Pastaba**
>
> If the algorithm does not need any additional user-set parameters (excluding the usual input and output layer parameters), then the algorithm is run immediately without any dialog popup.

   1. If parameters other than the usual input or output layers are needed, the algorithm dialog pops up. Fill in the required information.
   2. Click *Modify Selected Features* or *Modify All Features* depending on whether there's an active selection.

   Changes are applied to the layer and placed in the edit buffer: the layer is indeed toggled to editing mode with unsaved modification as indicated by the icon next to the layer name.

5. As usual, press Save layer edits to commit the changes in the layer. You can also press Undo to rollback the whole modification.

12.4 Skirtingų sluoksnių duomenų jungimas ir redagavimas

Galimybė jungti duomenis iš skirtingų sluoksnių yra viena iš GIS programinės įrangos pareigų. Tokia jungtis gali remtis erdviniu ryšiu tarp geoobjektų arba jų bendrais atributais. QGIS teikia įrankius, kurie dirba su abiem šiais ryšiais, tai yra:

- Apdorojimo algoritmai gali kurti naują sluoksnį, kaip jungimo rezultatą, konkrečiai *Join attributes by location*, *Join attributes by nearest*, *Join attributes by field value*, …
- SQL užklausos gali kurti naują sluoksnį iš *DB tvarkyklės* arba kaip *virtualų sluoksnį*
- *Jungimo savybės* arba *ryšių nustatymai*, kurie laikinai praplečia geoobjektų atributus nurodytame sluoksnyje tais atributais, kuriuos turi geoobjektai kitame sluoksnyje pagal kažkokį atitinkantį atributą(us).

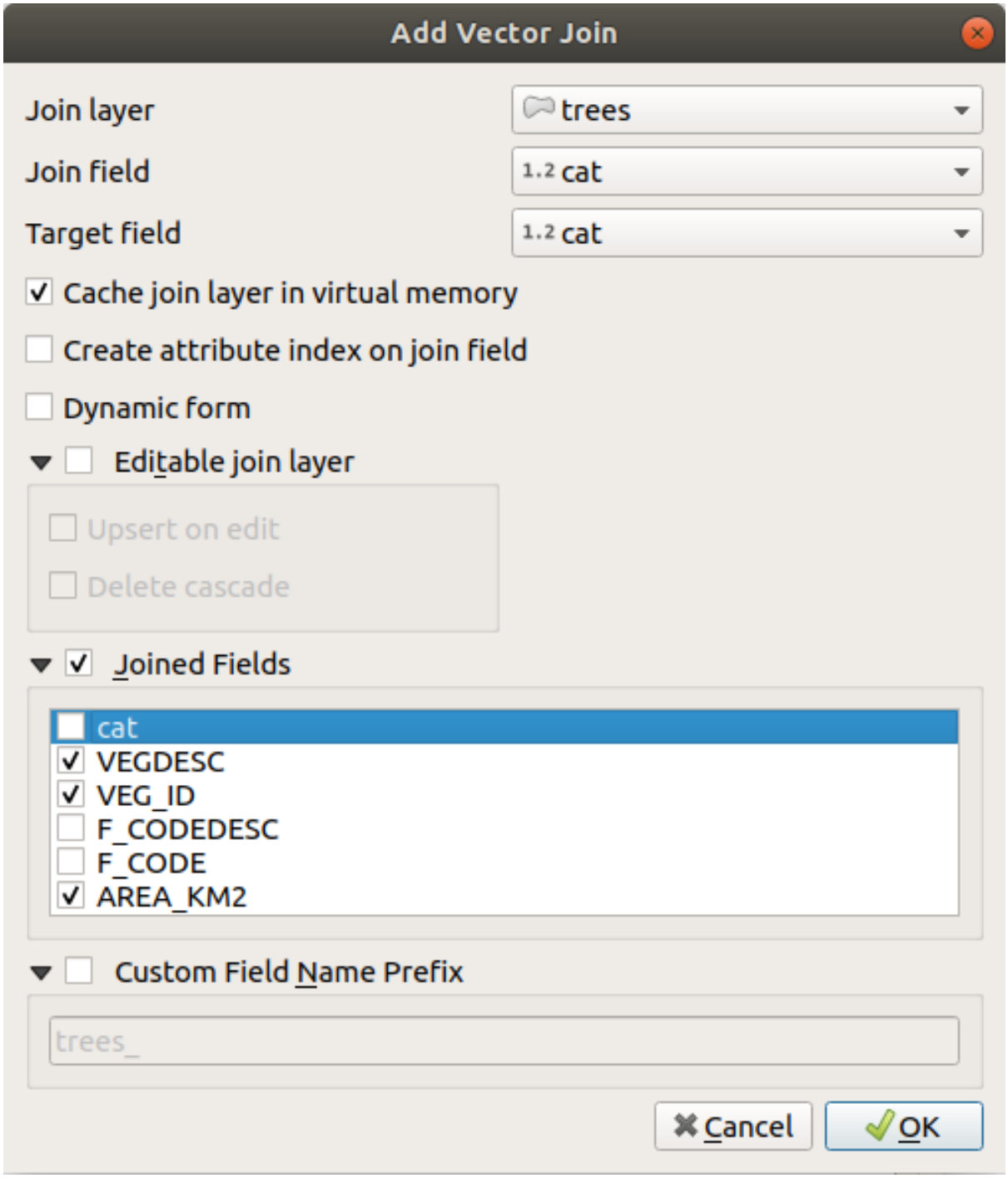
  Jungimai ir ryšiai yra techniniai terminai, pasiskolinti iš duomenų bazių, siekiant maksimalios naudos iš duomenų, saugomų lentelėse jungiant jų turinį. Mintis yra geoobjektus (įrašus) skirtinguose sluoksniuose (lentelėse) galima susieti vieną su kitu. Skaičius įrašų, kurie atitinka vienas kitą gali būti skirtingas (nulis, vienas, daug).

12.4.1 Dviejų sluoksnių geoobjektų sujungimas

QGIS **Jungtys** leidžia susieti dabartinio sluoksnio geoobjektus su kito įkelto vektorinio sluoksnio geoobjektais. Nepriklausomai nuo to, ar jie turi erdvinį komponentą ir kokio tipo yra jų geometrija. Jungtis remiasi atributu, kuris yra bendras abiems sluoksniams vienas-su-vienu jungtyje.

Kad sukurtumėte sluoksnio jungtį (žemiau identifikuojamą kaip `paskirties sluoksnis`):

1. Eikite į sluoksnio kortelę *Savybės* ► *Jungtys*
2. Spauskite mygtuką Pridėti naują jungtį. Pasirodo dialogas *Pridėti vektorinę jungtį*.
3. Parinkti *Sujungti sluoksnį*, kurį norite sujungti su paskirties vektoriniu sluoksniu
4. Nurodykite *Sujungti lauką* (iš `sujungiamo sluoksnio`) ir *Paskirties lauką* (iš `paskirties sluoksnio`). Tai laukai, kurie naudojami rasti atitinkančius geoobjektus abiejuose sluoksniuose, taigi jie rodo susijusias reikšmes.
5. Spauskite *Gerai*, parinktų parametrų santrauka pridedama į skydelį *Jungtys*.



12.104 Fig. : Atributų lentelės sujungimas su esamu vektoriniu sluoksniu

Aukščiau nurodyti sluoksniai sukurs jungtį, kur **VISI** atributai iš **pirmo atitinkančio geoobjekto** sujungiamame sluoksnyje pridedami prie paskirties sluoksnio geoobjekto. Geoobjektų sujungimo metu naudojama tokia logika:

- Grąžinami visi paskirties sluoksnio geoobjektai, nepriklausomai nuo to, ar jie turi atitikmenį
- Jei paskirties lauke yra pasikartojančios reikšmės, šiems geoobjektams priskiriamas tas pats geoobjektas iš sujungiamo sluoksnio.
- Jei sujungiamame lauke yra pasikartojančios atitinkančios reikšmės, parenkamas tik pirmas atitinkantis geoobjektas.

Pastaba

QGIS jungtys remiasi vieno lauko atitikimu, taigi dažniausiai jūs norite įsitikinti, kad atitinkamų laukų reikšmės būtų unikalios.

QGIS provides some more options to tweak the join:

- *Cache join layer in virtual memory*: allows you to cache values in memory (without geometries) from the joined layer in order to speed up lookups.
- *Create attribute index on the join field* to speed up lookups
- *Dynamic form*: helps to synchronize join fields on the fly, according to the *Target field*. This way, constraints for join fields are also correctly updated. Note that it's deactivated by default because it may be very time consuming if you have a lot of features or a myriad of joins.
- If the target layer is editable, then some icons will be displayed in the attribute table next to fields, in order to inform about their status:
  - : the join layer is not configured to be editable. If you want to be able to edit join features from the target attribute table, then you have to check the option *Editable join layer*.
  - : the join layer is well configured to be editable, but its current status is read only.
  - : the join layer is editable, but synchronization mechanisms are not activated. If you want to automatically add a feature in the join layer when a feature is created in the target layer, then you have to check the option *Upsert on edit*. Symmetrically, the option *Delete cascade* may be activated if you want to automatically delete join features.
- *Joined fields*: instead of adding all the fields from the joined layer, you can specify a subset.
- *Custom field name prefix* for joined fields, in order to avoid name collision

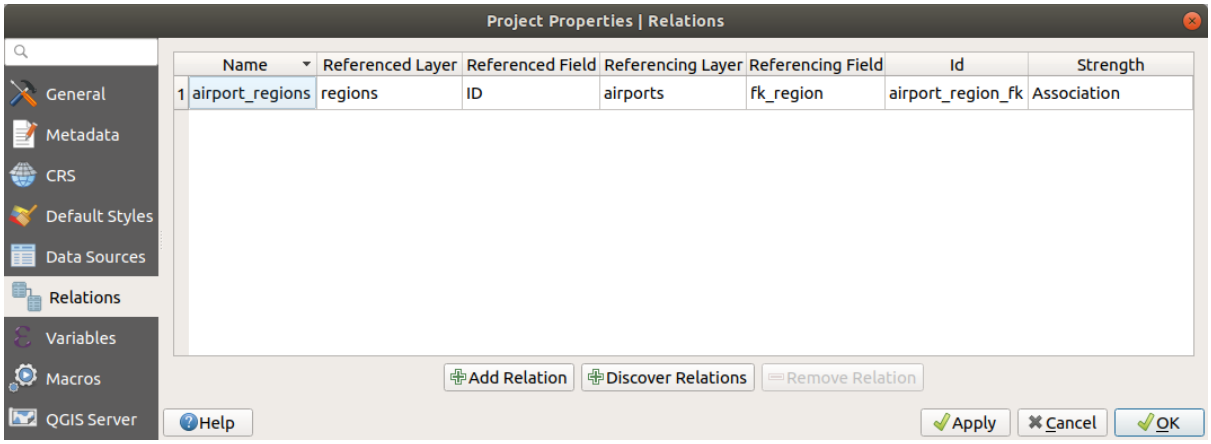
12.4.2 Setting relations between multiple layers

Unlike joins that define a one-to-one link between features across two layers, relations help you build interconnections between multiple features across two or more layers. As such, relations are project level settings and are set in *Project* ► *Properties* ► *Relations* tab. From there, you can:

- *Add relation* whose type can be:
  - *one to many relation*
  - *many to many relation*
  - *polymorphic relation* that you can add or edit with the dedicated tools in the action drop-down menu.

  Pastaba

  There is no simple way yet to edit a non-polymorphic relation once it has been created. Only the name can be edited with a double-click. For any other parameters of such a relation you will have to remove and recreate it.
- *Discover relations*: QGIS is able to discover existing relations from supported database formats (PostgreSQL, GeoPackage, ESRI File Geodatabase, …). This can be a good way to ease the relations definition.
- *Remove relation*

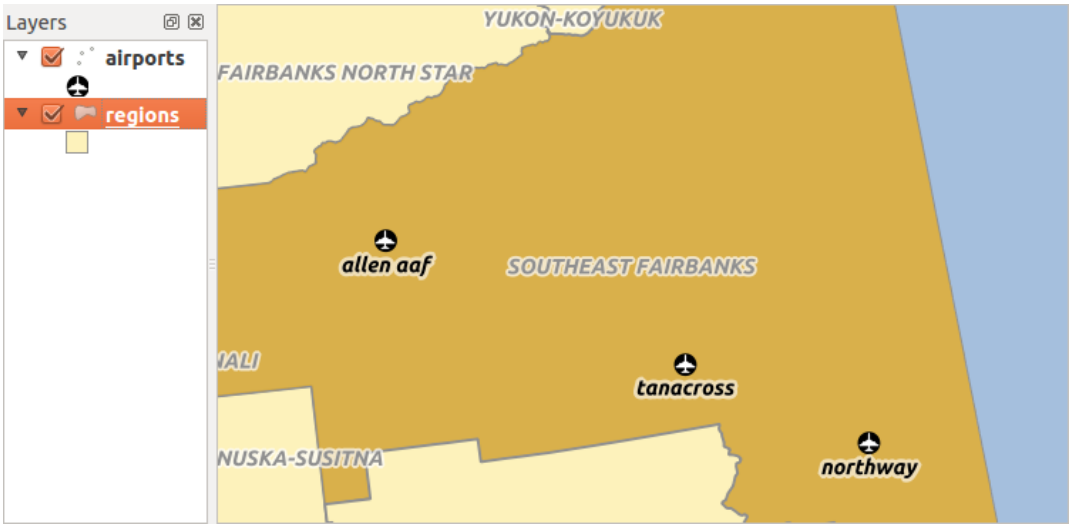


12.105 Fig. : Ryšių kortelė

One to many (1-N) relations

As an example you have a layer with all regions of Alaska (polygon) which provides some attributes about its name and region type and a unique id (which acts as primary key).

Then you get another point layer or table with information about airports that are located in the regions and you also want to keep track of these. If you want to add them to the regions layer, you need to create a one to many relation using foreign keys, because there are several airports in most regions.



12.106 Fig. : Alaska region with airports

Layers and keys

QGIS makes no difference between a table and a vector layer. Basically, a vector layer is a table with a geometry. So you can add your table as a vector layer. To demonstrate the 1-n relation, you can load the `regions` and `airports` layers in the sample dataset. In practice, each airport belongs to exactly one region while each region can have any number of airports (a typical one to many relation).

In addition to the attributes describing the airports, the aiports layer has another field `fk_region` which acts as a foreign key (if you have a database, you will probably want to define a constraint on it). This `fk_region` field will always contain an id of a region. It can be seen like a pointer to the region it belongs to.

All you have to do is to tell QGIS the relation between the layers so that you can design a custom edit form for editing and QGIS takes care of the setup. It works with different providers (so you can also use it with shape and csv files).

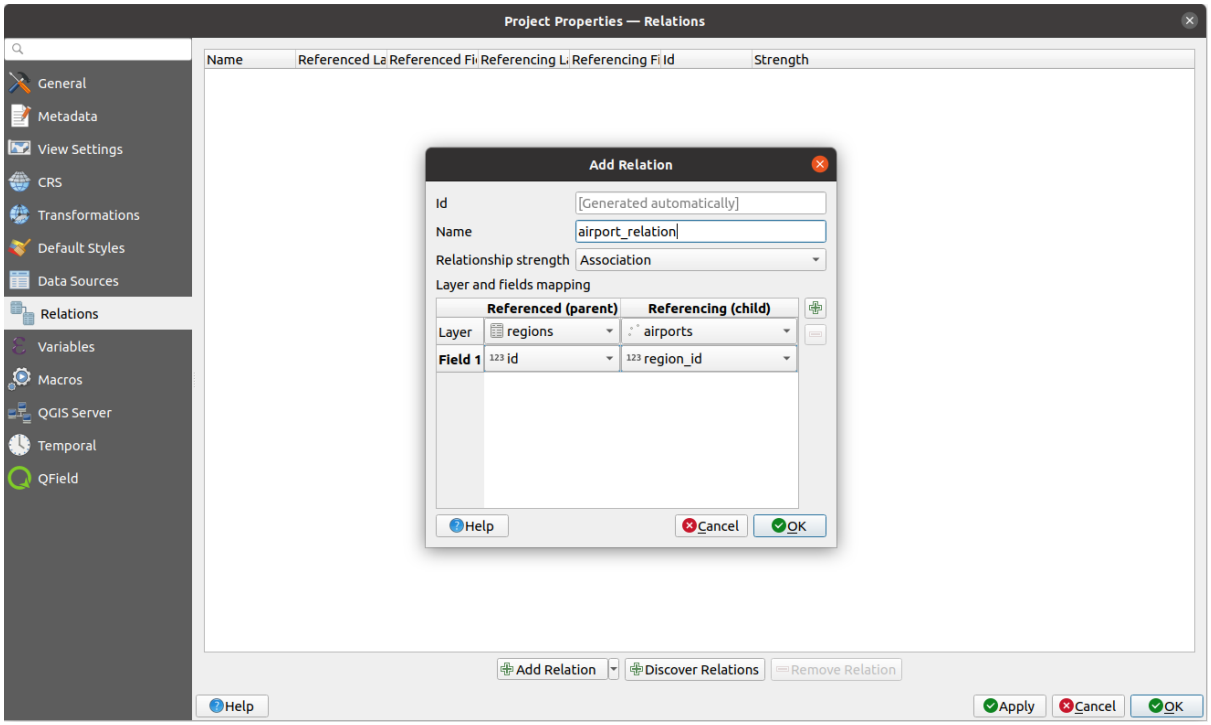
Defining 1-N relations

The first thing we are going to do is to let QGIS know about the relations between the layers. This is done in *Project* ► *Properties…*. Open the *Relations* tab and click on *Add Relation*.

- **Name** is going to be used as a title. It should be a human readable string describing what the relation is used for. We will just call say **airport_relation** in this case.
- **Referenced Layer (Parent)** also considered as parent layer, is the one with the primary key, pointed to, so here it is the `regions` layer. You need to define the primary key of the referenced layer, so it is `ID`.
- **Referencing Layer (Child)** also considered as child layer, is the one with the foreign key field on it. In our case, this is the `airports` layer. For this layer you need to add a referencing field which points to the other layer, so this is `fk_region`.

> **Pastaba**
>
> Sometimes, you need more than a single field to uniquely identify features in a layer. Creating a relation with such a layer requires a **composite key**, i.e. more than a single pair of matching fields. Use the Add new field pair as part of a composite foreign key button to add as many pairs as necessary.

- **Id** will be used for internal purposes and has to be unique. You may need it to build *custom forms*. If you leave it empty, one will be generated for you but you can assign one yourself to get one that is easier to handle
- **Relationship strength** sets the strength of the relation between the parent and the child layer. The default *Association* type means that the parent layer is *simply* linked to the child one while the *Composition* type allows you to duplicate also the child features when duplicating the parent ones and on deleting a feature the children are deleted as well, resulting in cascade over all levels (means children of children of… are deleted as well).

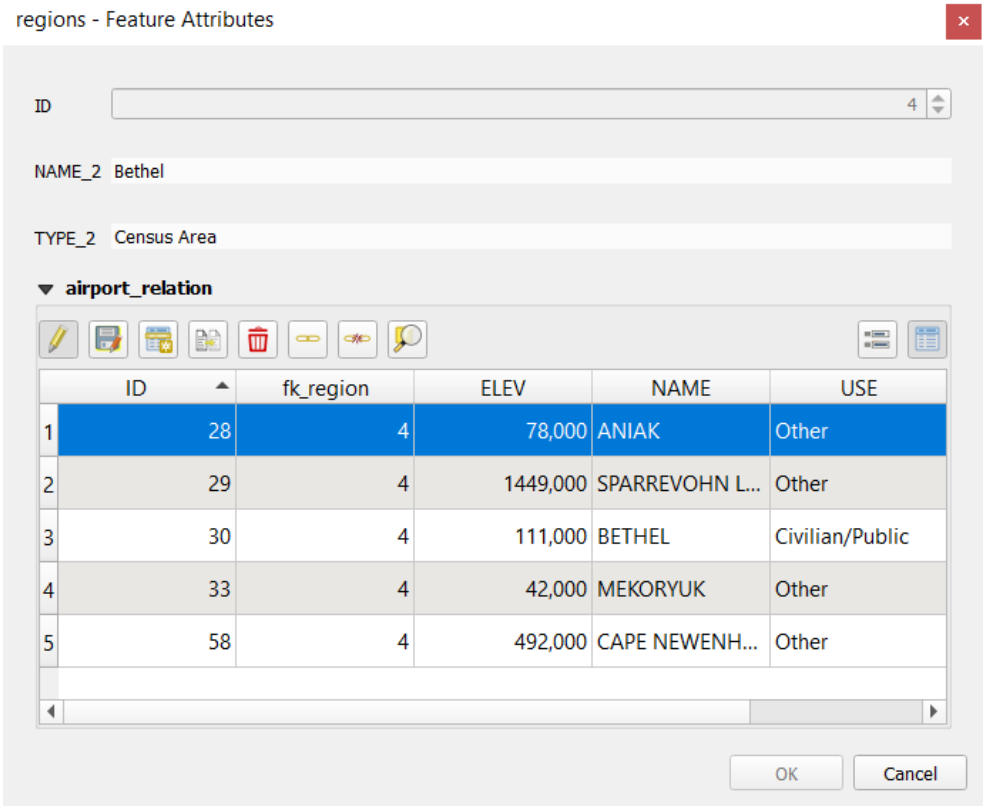


12.107 Fig. : Adding a relation between regions and airports layers

From the *Relations* tab, you can also press the *Discover Relation* button to fetch the relations available from the providers of the loaded layers. This is possible for layers stored in data providers like PostgreSQL or SpatiaLite.

Forms for 1-N relations

Now that QGIS knows about the relation, it will be used to improve the forms it generates. As we did not change the default form method (autogenerated), it will just add a new widget in our form. So let's select the layer region in the legend and use the identify tool. Depending on your settings, the form might open directly or you will have to choose to open it in the identification dialog under actions.



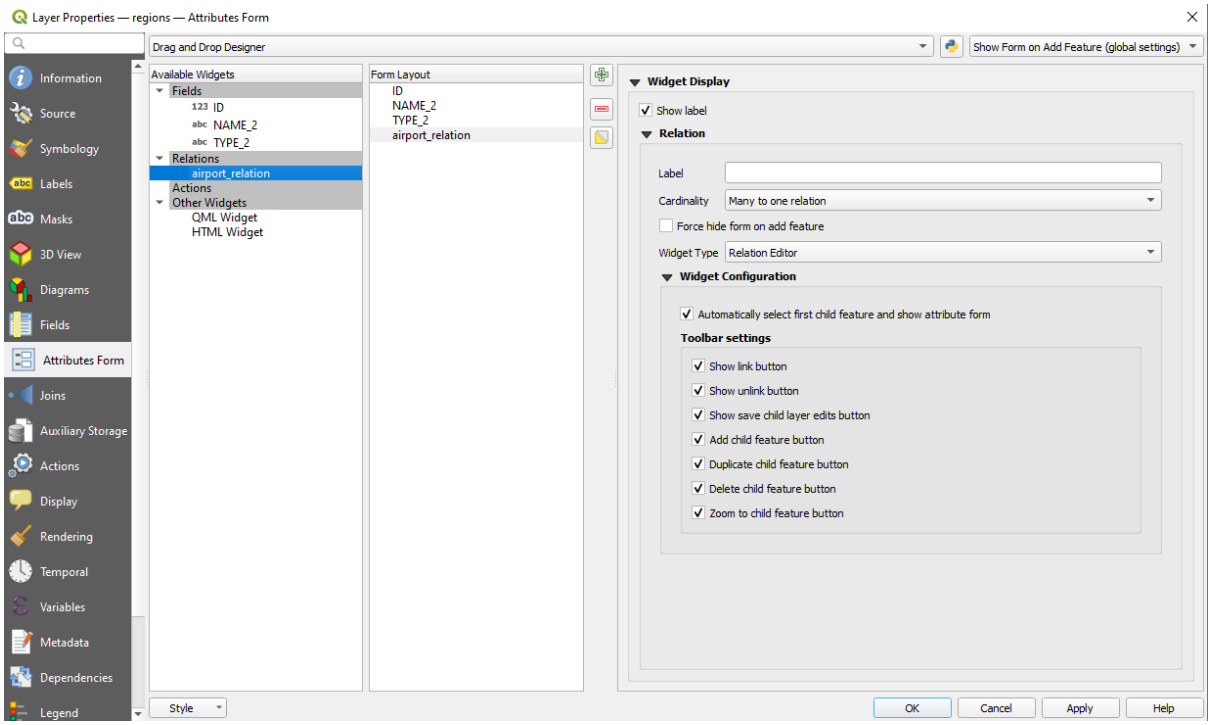
12.108 Fig. : Identification dialog regions with relation to airports

As you can see, the airports assigned to this particular region are all shown in a table. And there are also some buttons available. Let's review them shortly:

- The button is for toggling the edit mode. Be aware that it toggles the edit mode of the airport layer, although we are in the feature form of a feature from the region layer. But the table is representing features of the airport layer.
- The button is for saving all the edits in the child layer (airport).
- The button lets you digitize the airport geometry in the map canvas and assigns the new feature to the current region by default. Note that the icon will change according to the geometry type.
- The button adds a new record to the airport layer attribute table and assigns the new feature to the current region by default. The geometry can be drawn later with the *Add part* digitizing tool.
- The button allows you to copy and paste one or more child features within the child layer. They can later be assigned to a different parent feature or have their attributes modified.
- The button deletes the selected airport(s) permanently.
- The symbol opens a new dialog where you can select any existing airport which will then be assigned to the current region. This may be handy if you created the airport on the wrong region by accident.

- The symbol unlinks the selected airport(s) from the current region, leaving them unassigned (the foreign key is set to NULL) effectively.
- With the button you can zoom the map to the selected child features.
- The two buttons and to the right switch between the *table view and form view* of the related child features.

If you use the *Drag and Drop Designer* for the regions feature, you can select which tools are available. You can even decide whether to open a new form when a new feature is added using *Force hide form on add feature* option. Be aware that this option implies that not null attributes must take a valid default value to work correctly.



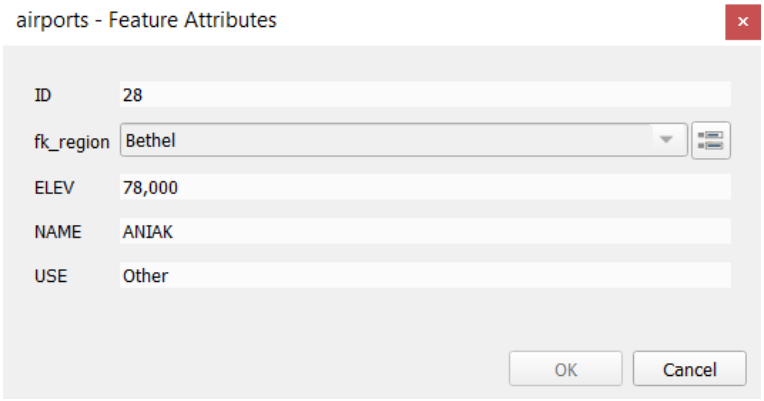
12.109 Fig. : Drag and Drop Designer for configure regions-airports relation tools

In the above example the referencing layer has geometries (so it isn't just an alphanumeric table) so the above steps will create an entry in the layer attribute table that has no corresponding geometric feature. To add the geometry:

1. Choose *Open Attribute Table* for the referencing layer.
2. Select the record that has been added previously within the feature form of the referenced layer.
3. Use the Add Part digitizing tool to attach a geometry to the selected attributes table record.

If you work on the airport table, the widget Relation Reference is automatically set up for the `fk_region` field (the one used to create the relation), see *Relation Reference widget*.

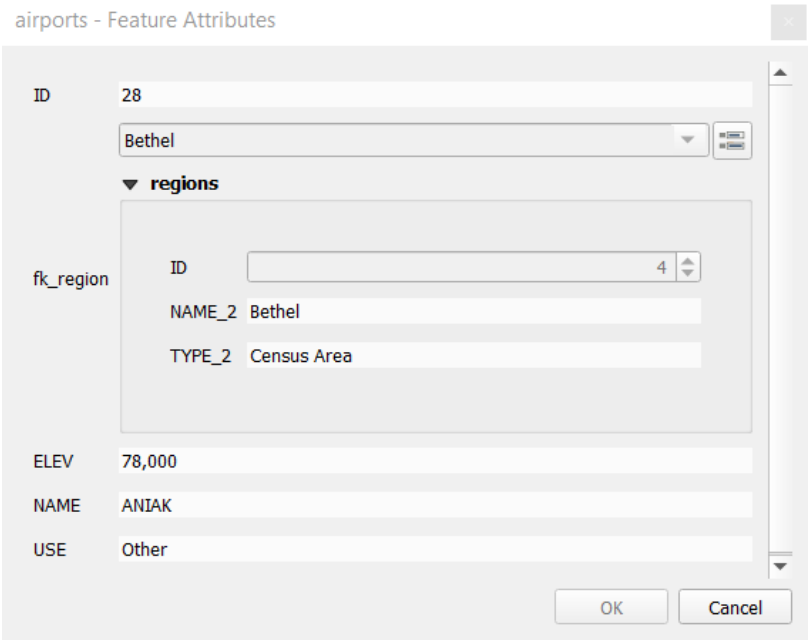
In the airport form you will see the button at the right side of the `fk_region` field: if you click on the button the form of the region layer will be opened. This widget allows you to easily and quickly open the forms of the linked parent features.



12.110 Fig. : Identification dialog airport with relation to regions

The Relation Reference widget has also an option to embed the form of the parent layer within the child one. It is available in the *Properties* ► *Attributes Form* menu of the airport layer: select the `fk_region` field and check the `Show embedded form` option.

If you look at the feature dialog now, you will see that the form of the region is embedded inside the airports form and will even have a combobox, which allows you to assign the current airport to another region.

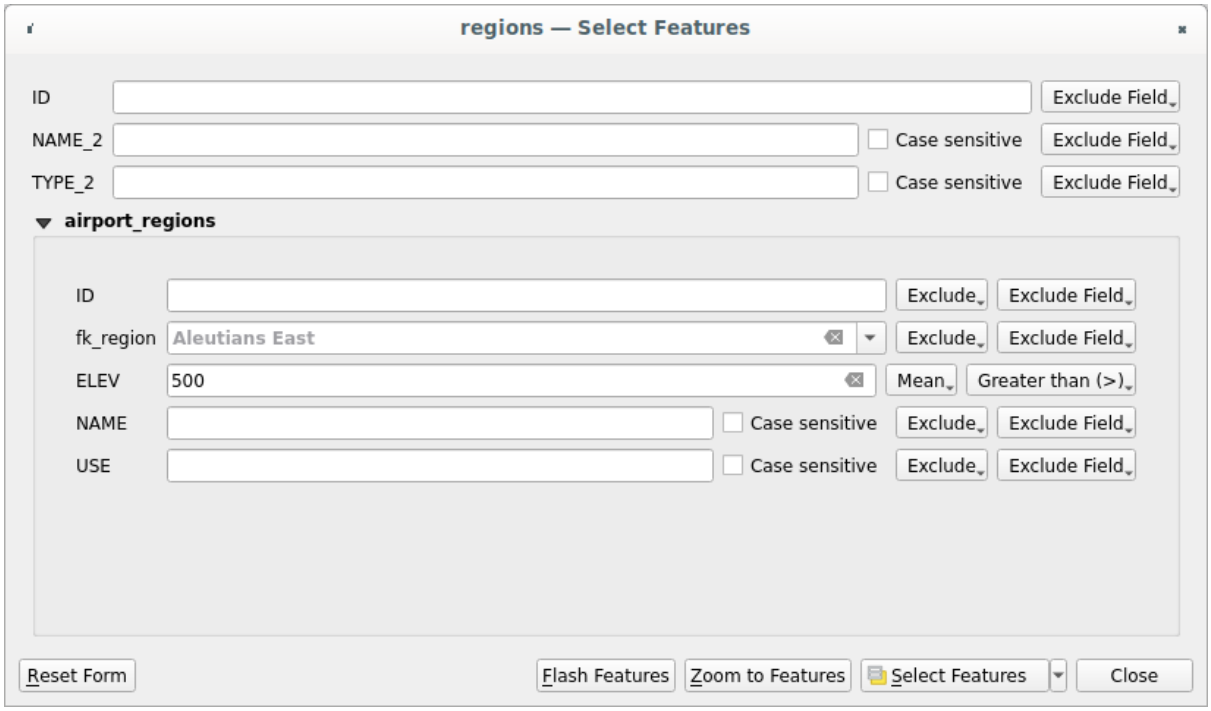


Moreover if you toggle the editing mode of the airport layer, the `fk_region` field has also an autocompleter function: while typing you will see all the values of the `id` field of the region layer. Here it is possible to digitize a polygon for the region layer using the button if you chose the option `Allow adding new features` in the *Properties* ► *Attributes Form* menu of the airport layer.

The child layer can also be used in the *Parinkti geoobjektus pagal reikšmę* tool in order to select features of the parent layer based on attributes of their children.

In Fig. 12.111, all the regions where the mean altitude of the airports is greater than 500 meters above sea level are selected.

You will find that many different aggregation functions are available in the form.



12.111 Fig. : Select parent features with child values

Many-to-many (N-M) relations

N-M relations are many-to-many relations between two tables. For instance, the `airports` and `airlines` layers: an airport receives several airline companies and an airline company flies to several airports.

This SQL code creates the three tables we need for an N-M relationship in a PostgreSQL schema named *locations*. You can run the code using the *Database* ► *DB Manager*… for PostgreSQL or external tools such as pgAdmin. The airports table stores the `airports` layer and the airline table stores the `airlines` layer. In both tables few fields are used for clarity. The *tricky* part is the `airports_airlines` table. We need it to list all airlines for all airports (or vice versa). This kind of table is known as a *pivot table*. The *constraints* in this table force that an airport can be associated with an airline only if both already exist in their layers.

```
CREATE SCHEMA locations;

CREATE TABLE locations.airports
(
   id serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   airport_name text NOT NULL,
   CONSTRAINT airports_pkey PRIMARY KEY (id)
);

CREATE INDEX airports_geom_idx ON locations.airports USING gist (geom);

CREATE TABLE locations.airlines
(
   id serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   airline_name text NOT NULL,
   CONSTRAINT airlines_pkey PRIMARY KEY (id)
);

CREATE INDEX airlines_geom_idx ON locations.airlines USING gist (geom);
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
CREATE TABLE locations.airports_airlines
(
   id serial NOT NULL,
   airport_fk integer NOT NULL,
   airline_fk integer NOT NULL,
   CONSTRAINT airports_airlines_pkey PRIMARY KEY (id),
   CONSTRAINT airports_airlines_airport_fk_fkey FOREIGN KEY (airport_fk)
      REFERENCES locations.airports (id)
      ON DELETE CASCADE
      ON UPDATE CASCADE
      DEFERRABLE INITIALLY DEFERRED,
   CONSTRAINT airports_airlines_airline_fk_fkey FOREIGN KEY (airline_fk)
      REFERENCES locations.airlines (id)
      ON DELETE CASCADE
      ON UPDATE CASCADE
      DEFERRABLE INITIALLY DEFERRED
 );
```

Instead of PostgreSQL you can also use GeoPackage. In this case, the three tables can be created manually using the *Database* ► *DB Manager*…. In GeoPackage there are no schemas so the *locations* prefix is not needed.

Foreign key constraints in `airports_airlines` table can´t be created using *Table* ► *Create Table…* or *Table* ► *Edit Table…* so they should be created using *Database* ► *SQL Window*…. GeoPackage doesn't support *ADD CONSTRAINT* statements so the `airports_airlines` table should be created in two steps:

1. Set up the table only with the `id` field using *Table* ► *Create Table…*
2. Using *Database* ► *SQL Window…*, type and execute this SQL code:

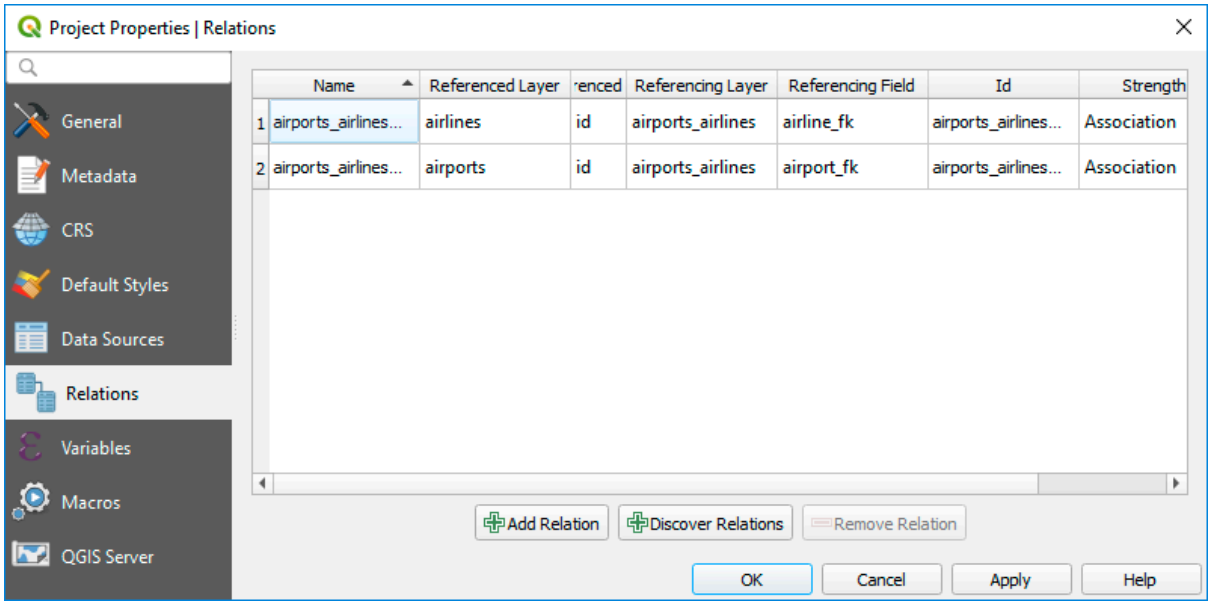
```
ALTER TABLE airports_airlines
   ADD COLUMN airport_fk INTEGER
   REFERENCES airports (id)
   ON DELETE CASCADE
   ON UPDATE CASCADE
   DEFERRABLE INITIALLY DEFERRED;

ALTER TABLE airports_airlines
   ADD COLUMN airline_fk INTEGER
   REFERENCES airlines (id)
   ON DELETE CASCADE
   ON UPDATE CASCADE
   DEFERRABLE INITIALLY DEFERRED;
```

Then in QGIS, you should set up two *one-to-many relations* as explained above:

- a relation between `airlines` table and the pivot table;
- and a second one between `airports` table and the pivot table.

An easier way to do it (only for PostgreSQL) is using the *Discover Relations* in *Project* ► *Properties* ► *Relations*. QGIS will automatically read all relations in your database and you only have to select the two you need. Remember to load the three tables in the QGIS project first.



12.112 Fig. : Relations and autodiscover

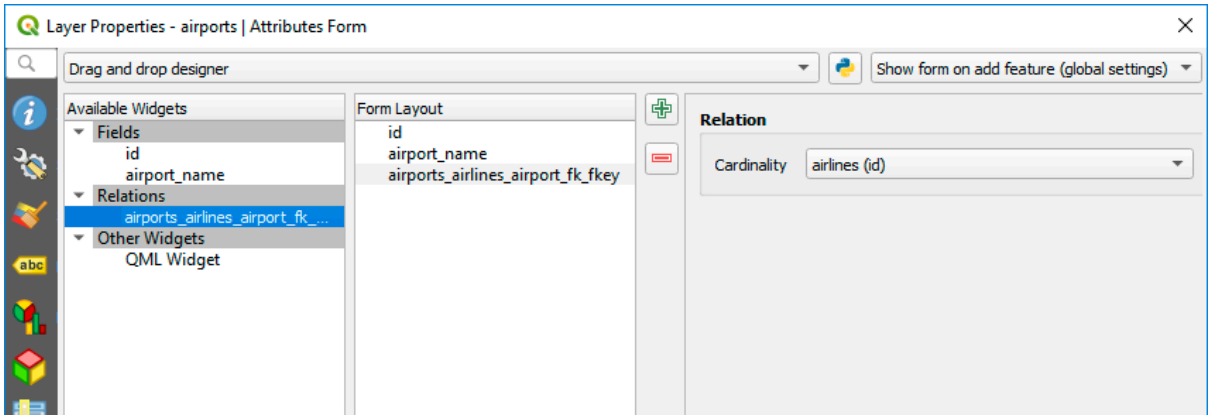
In case you want to remove an `airport` or an `airline`, QGIS won't remove the associated record(s) in `airports_airlines` table. This task will be made by the database if we specify the right *constraints* in the pivot table creation as in the current example.

Pastaba

Combining N-M relation with automatic transaction group

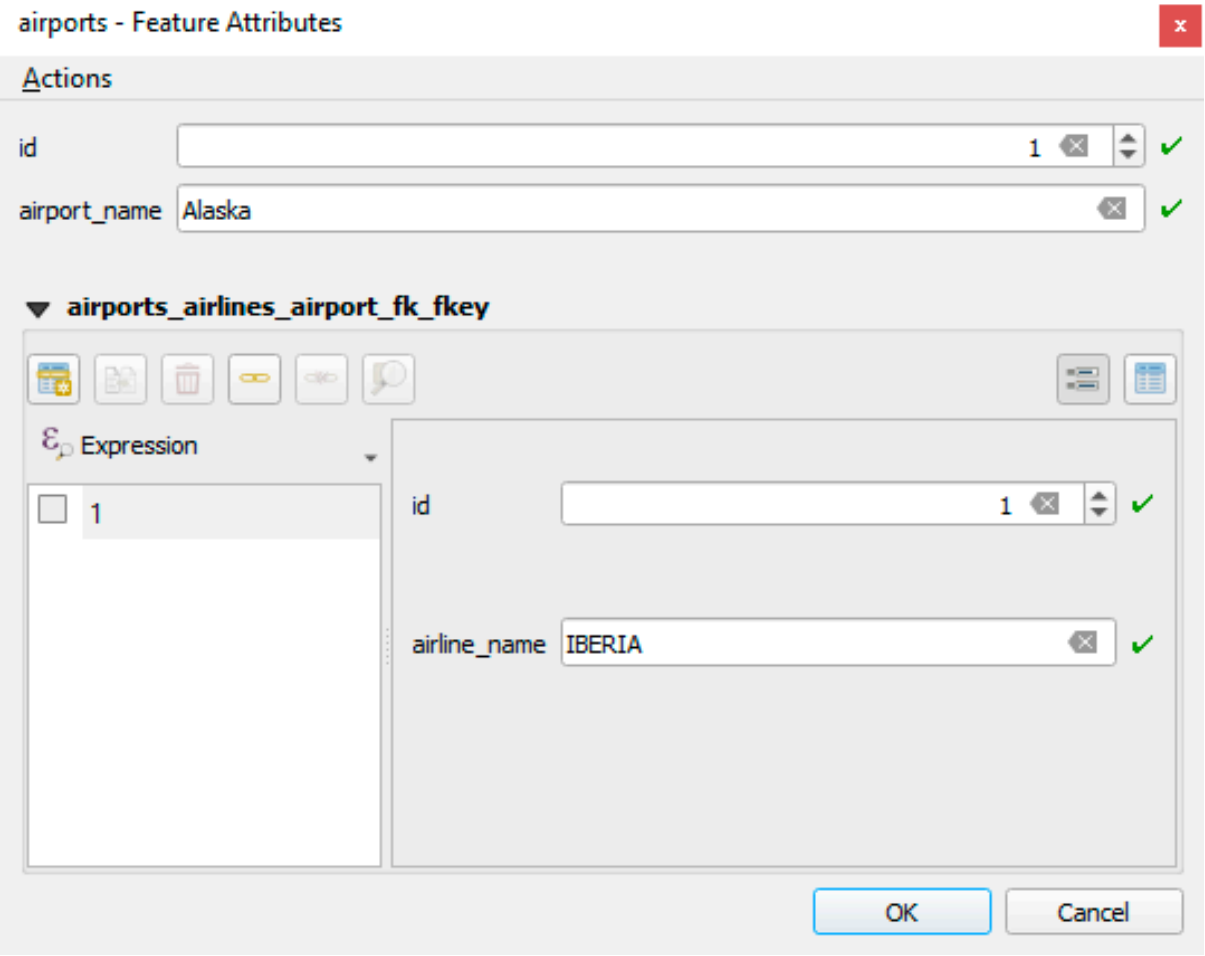
You should enable the transaction mode in *Project Properties* ► *Data Sources* ► when working on such context. QGIS should be able to add or update row(s) in all tables (airlines, airports and the pivot tables).

Finally we have to select the right cardinality in the *Layer Properties* ► *Attributes Form* for the `airports` and `airlines` layers. For the first one we should choose the **airlines (id)** option and for the second one the **airports (id)** option.



12.113 Fig. : Set relationship cardinality

Now you can associate an airport with an airline (or an airline with an airport) using *Add child feature* or *Link existing child feature* in the subforms. A record will automatically be inserted in the `airports_airlines` table.



12.114 Fig. : N-M relationship between airports and airlines

Pastaba

Using **Many to one relation** cardinality

Sometimes hiding the pivot table in an N-M relationship is not desirable. Mainly because there are attributes in the relationship that can only have values when a relationship is established. If your tables have a geometry field, it could be interesting to activate the *On map identification* option (*Layer Properties* ► *Attributes Form* ► *Available widgets* ► *Fields*) for the foreign key fields in the pivot table.

Pastaba

Pivot table primary key

Avoid using multiple fields in the primary key in a pivot table. QGIS assumes a single primary key so a constraint like `constraint airports_airlines_pkey primary key (airport_fk, airline_fk)` will not work.

Polymorphic relations

The purpose

Polymorphic relations are special case of 1-N relations, where a single referencing (document) layer contains the features for multiple referenced layers. This differs from normal relations which require different referencing layer for each referenced layer. A single referencing (document) layer is achieved by adding an adiditonal `layer_field` column in the referencing (document) layer that stores information to identify the referenced layer. In its most simple form, the referencing (document) layer will just insert the layer name of the referenced layer into this field.

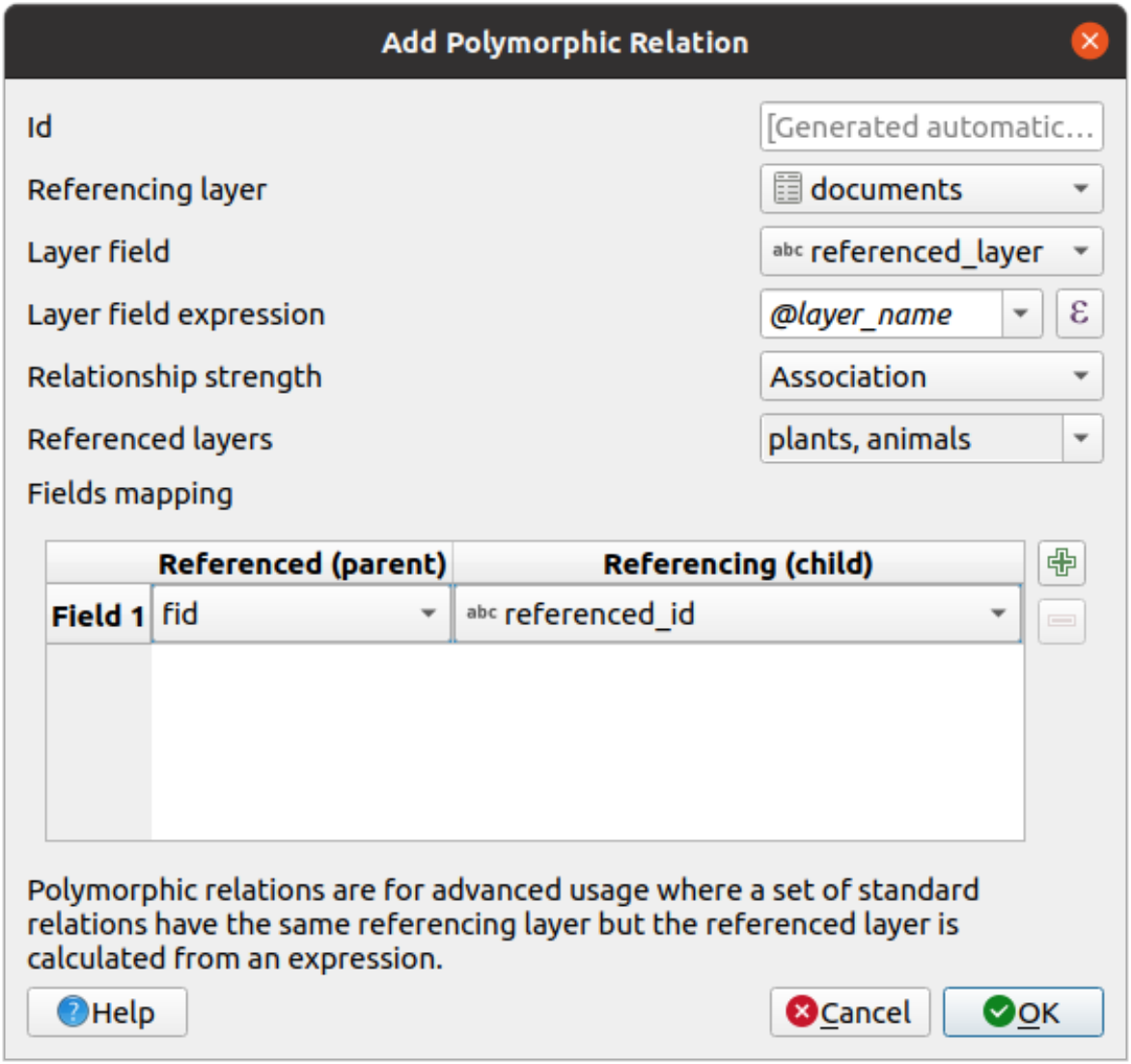
To be more precise, a polymorphic relation is a set of normal relations having the same referencing layer but having the referenced layer dynamically defined. The polymorphic setting of the layer is solved by using an expression which has to match some properties of the referenced layer like the table name, layer id, layer name.

Imagine we are going to the park and want to take pictures of different species of `plants` and `animals` we see there. Each plant or animal has multiple pictures associated with it, so if we use the normal 1:N relations to store pictures, we would need two separate tables, `animal_images` and `plant_images`. This might not be a problem for 2 tables, but imagine if we want to take separate pictures for mushrooms, birds etc.

Polymorphic relations solve this problem as all the referencing features are stored in the same table `documents`. For each feature the referenced layer is stored in the `referenced_layer` field and the referenced feature id in the `referenced_id` field.

Defining polymorphic relations

First, let QGIS know about the polymorphic relations between the layers. This is done in *Project* ► *Properties…*. Open the *Relations* tab and click on the little down arrow next to the *Add Relation* button, so you can select the *Add Polymorphic Relation* option from the newly appeared dropdown.



12.115 Fig. : Adding a polymorphic relation using `documents` layer as referencing and `animals` and `plants` as referenced layers.

- **Id** will be used for internal purposes and has to be unique. You may need it to build *custom forms*. If you leave it empty, one will be generated for you but you can assign one yourself to get one that is easier to handle
- **Referencing Layer (Child)** also considered as child layer, is the one with the foreign key field on it. In our case, this is the `documents` layer. For this layer you need to add a referencing field which points to the other layer, so this is `referenced_id`.

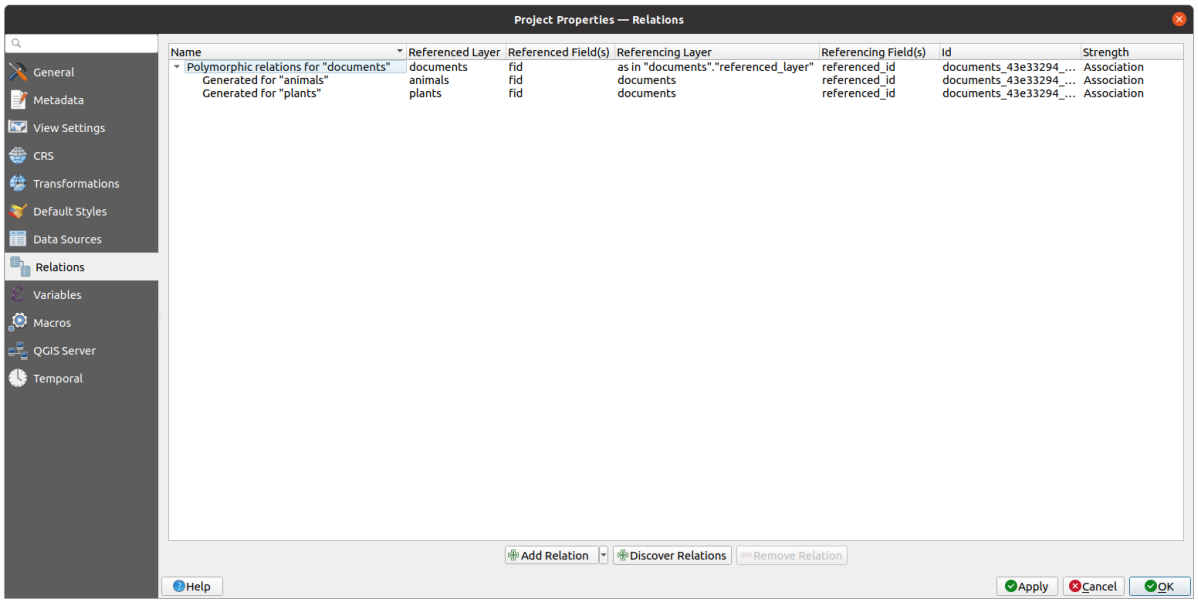
> **Pastaba**
>
> Sometimes, you need more than a single field to uniquely identify features in a layer. Creating a relation with such a layer requires a **composite key**, i.e. more than a single pair of matching fields. Use the Add new field pair as part of a composite foreign key button to add as many pairs as necessary.

- **Layer Field** is the field in the referencing table that stores the result of the evaluated layer expression which is the referencing table that this feature belongs to. In our example, this would be the `referenced_layer` field.
- **Layer expression** evaluates to a unique identifier of the layer. This can be the layer name `@layer_name`,

the layer id `@layer_id`, the layer's table name `decode_uri(@layer, 'table')` or anything that can uniquely identifies a layer.

- **Relationship strength** sets the strength of the generated relations between the parent and the child layer. The default *Association* type means that the parent layer is *simply* linked to the child one while the *Composition* type allows you to duplicate also the child features when duplicating the parent ones and on deleting a feature the children are deleted as well, resulting in cascade over all levels (means children of children of… are deleted as well).
- **Referenced Layers** also considered as parent layers, are those with the primary key, pointed to, so here they would be `plants` and `animals` layers. You need to define the primary key of the referenced layers from the dropdown, so it is `fid`. Note that the definition of a valid primary key requires all the referenced layers to have a field with that name. If there is no such field you cannot save a polymorphic relation.

Once added, the polymorphic relation can be edited via the *Edit Polymorphic Relation* menu entry.



12.116 Fig. : Preview of the newly created polymorphic relation and its child relations for animals and plants.

The example above uses the following database schema:

```
CREATE SCHEMA park;

CREATE TABLE park.animals
(
   fid serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   animal_species text NOT NULL,
   CONSTRAINT animals_pkey PRIMARY KEY (fid)
);

CREATE INDEX animals_geom_idx ON park.animals USING gist (geom);

CREATE TABLE park.plants
(
   fid serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   plant_species text NOT NULL,
   CONSTRAINT plants_pkey PRIMARY KEY (fid)
);
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
CREATE INDEX plants_geom_idx ON park.plants USING gist (geom);

CREATE TABLE park.documents
(
   fid serial NOT NULL,
   referenced_layer text NOT NULL,
   referenced_id integer NOT NULL,
   image_filename text NOT NULL,
   CONSTRAINT documents_pkey PRIMARY KEY (fid)
);
```

Once relations are established, the *Identify Features* result panel will display related features, allowing you to view associated data directly.

Dataset-stored Relationships

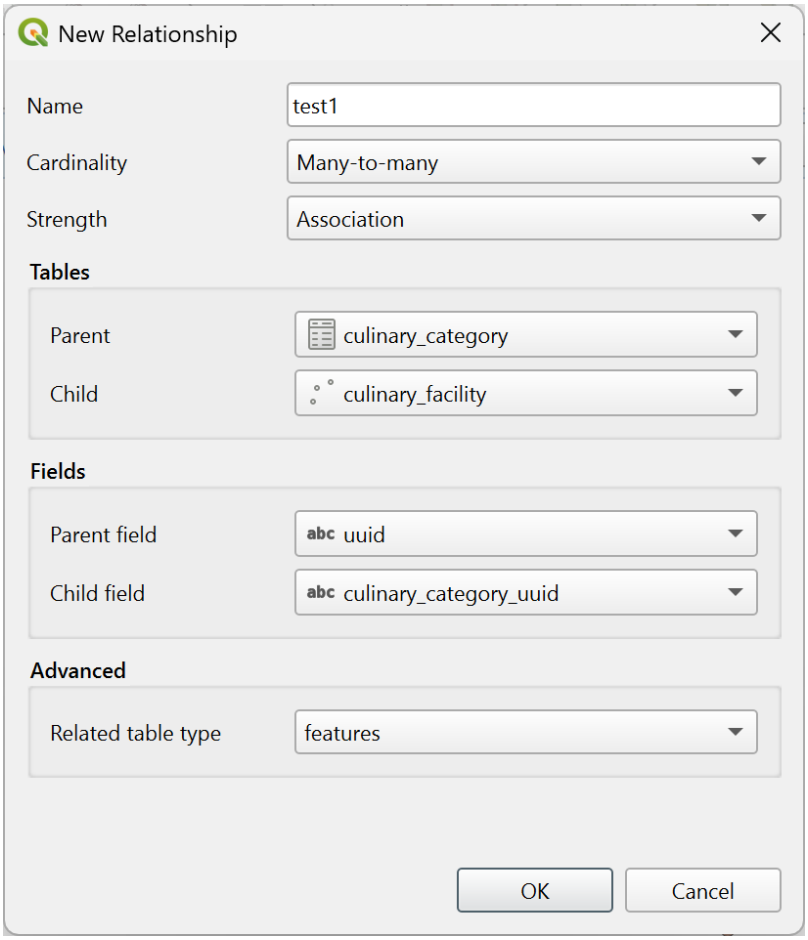
Other than the above mentioned relationships that are created in the project properties and apply only in the active project, QGIS can also read database-level relations, i.e., connections defined between layers in the dataset. This is supported for database providers such as PostgreSQL, SpatiaLite, GeoPackage, ESRI FileGeodatabase. When such parent and child layers are loaded in a project, QGIS can automatically detect the relation settings and fill appropriately the *Project* ► *Project properties* ► *Relations* tab. This can also be triggered using the *Discover relations* button.

For some data types (e.g., GeoPackage, SpatiaLite or ESRI FileGeodatabase), QGIS also allows to create and edit the relations between the tables in the database. This can be managed through the *Browser Panel*. From the *Browser panel*, right-click on the database entry, select *New relationship…*, and in the new dialog, fill the following options:

- *Name*: set the name of the new relationship.
- *Cardinality*: define the cardinality of the relationship.
- *Strength*: specify the strength of the relationship.
- Under the *Tables* group, set your *Parent* and *Child* tables.
- Under the *Fields* group, set your *Parent field* and *Child field*.
- Under the *Advanced* section, depending on the database type, you can set:
  - *Related table type*:
    - *features*: for standard feature tables.
    - *media*: for tables storing media files.
    - *simple_attributes*: for tables with simple attribute data.
    - *attributes*: for tables with complex attribute data.
    - *tiles*: for tables storing tile data.
  - *Forward label*: a name to uniquely identify the relationship when navigating from the parent table to the child table.
  - *Reverse label*: a name to uniquely identify the relationship when navigating from the child table to the parent table.

Applying the dialog will add a *Relationships* node under the database entry, with the new relationship as child item.

Editing or deleting existing relationships can be done by right-clicking on the relationship in the Browser Panel.



12.117 Fig. : Managing dataset-stored relationships in the Browser Panel

12.4.3 Storing and fetching an external resource

A field may target a resource stored on an external storage system. Attribute forms can be configured so they act as a client to an external storage system in order to store and fetch those resources, on users demand, directly from the forms.

Configuring an external storage

In order to setup an external storage, you have to first configure it from the vector *attribute form properties* and select the *Attachment* widget.

12.118 Fig. : Editing a WebDAV external storage for a given field

From the *Attachment* widget, you have to first select the *Storage type*:

- *Select Existing File*: The target URL already exists. When you select a resource, no store operation is achieved, the attribute is simply updated with the URL.
- *Simple Copy*: Stores a copy of the resource on a file disk destination (which could be a local or network shared file system) and the attribute is updated with the path to the copy.
- *WebDAV Storage*: The resource is pushed to a HTTP server supporting the WebDAV protocol and the attribute is updated with its URL. Nextcloud, Pydio or other file hosting software support this protocol.
- *AWS S3*: The resource is pushed to a server supporting AWS Simple Storage Service protocol and the attribute is updated with its URL. Amazon Web Service and MinIO hosting software support this protocol.

Then, you have to set up the *Store URL* parameter, which provides the URL to be used when a new resource needs to be stored. It's possible to set up an expression using the *data defined override widget* in order to have specific values according to feature attributes.

The variable **@selected_file_path** could be used in that expression and represent the absolute file path of the user selected file (using the file selector or drag'n drop).

Pastaba

Using the **WebDAV** or **AWS S3** external storage, if the URL ends with a „/", it is considered as a folder and the selected file name will be appended to get the final URL.

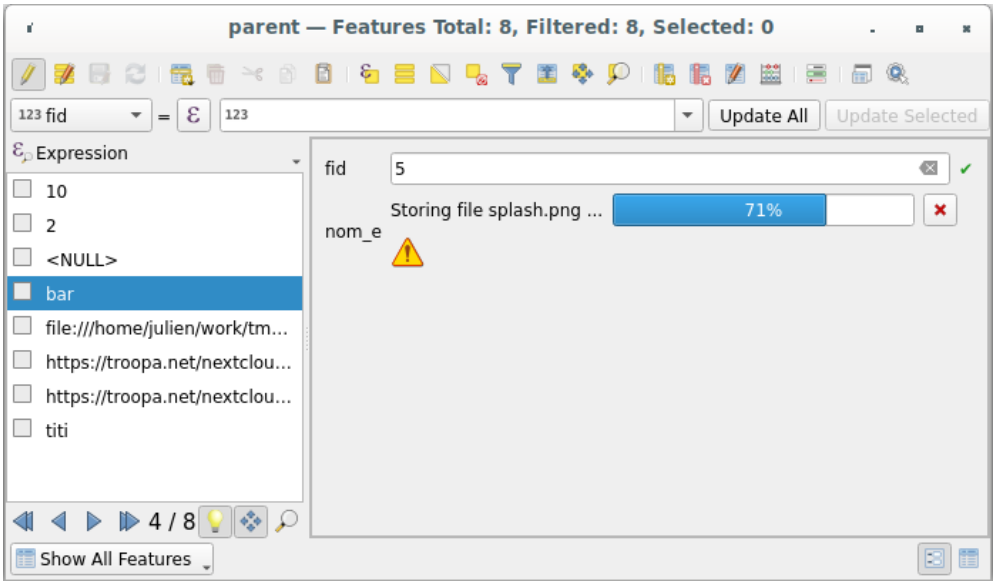
If the external storage system needs to, it's possible to configure an *authentication*.

Pastaba

To use the **AWS S3** external storage, you must use an **AWS S3** authentication type.

Using an external storage

Once configured, you can select a local file using the button … when editing a feature's attribute. Depending on the configured *storage type*, the file will be stored on the external storage system (except if *Select existing file* has been selected) and the field will be updated with the new resource URL.



12.119 Fig. : Storing a file to a WebDAV external storage

Pastaba

User can also achieve the same result if he drags and drops a file on the whole attachment widget.

Use the Cancel button to abort the storing process. It's possible to configure a viewer using the *Integrated document viewer* so the resource will be automatically fetched from the external storage system and displayed directly below the URL. The above icon indicates that the resource cannot be fetched from the external storage system. In that case, more details might appear in the *Pranešimų žurnalo skydelis*.

SKYRIUS 13

Working with Raster Data

13.1 Rastro savybių dialogas

Rastro duomenys sudaryti iš pikselių (arba celių), kur kiekvienas pikselis turi reikšmę. Paprastai jame saugomi skirtingi duomenų tipai, įskaitant:

- Vaizdai, tokie kaip palydovų vaizdai, skaitmeninės aero nuotraukos, skenuoti žemėlapiai
- Aukščio duomenys, tokie kaip skaitmeniniai aukščio modeliai (DEM), skaitmeniniai paviršiaus modeliai (DTM)
- Kiti duomenų tipai, tokie kaip žemės danga, dirvožemio tipai, krituliai ir daug kitų.

Raster data can be stored in several supported formats, including GeoTIFF, ERDAS Imagine, ArcInfo ASCII GRID, PostgreSQL Raster and others. See more at *Duomenų atidarymas*.

Norėdami pažiūrėti ar keisti rastro sluoksnio savybes, du kartus spauskite sluoksnio pavadinimą žemėlapio legendoje arba dešiniu pelės mygtuku spauskite sluoksnio pavadinimą ir kontekstiniame meniu parinkite *Savybės*. Tai atvers dialogą *Rastro sluoksnio savybės*.

Dialoge yra kelios kortelės:

Informacija	*Šaltinis*	*Symbology* ([1])
Transparency ([1])	*Labels* ([1])	*Histogram* ([1])
Braižymas	*Laikas*	*Piramidės*
Aukštis	*Metaduomenys*	*Legend*
Rodymas	*Attribute Tables*	*QGIS Serveris*
External plugins ([2]) tabs		

[1] Also available in the *Layer styling panel*
[2] *External plugins* you install can optionally add tabs to this dialog. Those are not presented in this document. Refer to their documentation.

Patarimas

Automatinio atnaujinimo braižymas

Sluoksnio stiliaus skydelis teikia kelias bendras sluoksnio savybių dialogo savybes ir yra geras valdiklis, kuris gali pagreitinti sluoksnio stiliaus konfigūravimą iš karto matydami jūsų pakeitimus žemėlapio drobėje.

Pastaba

Kadangi įtrauktų sluoksnių (žr. *Išorinių projektų sluoksnių įtraukimas*) savybės (simbologija, užrašai, veiksmai, numatytos reikšmės, formos…) traukiamos iš originalaus projekto failo, kad būtų išvengta pakeitimų, kurie gali sulaužyti šią elgseną, sluoksnio savybių dialogas šiems sluoksniams neprieinamas.

13.1.1 Informacinės savybės

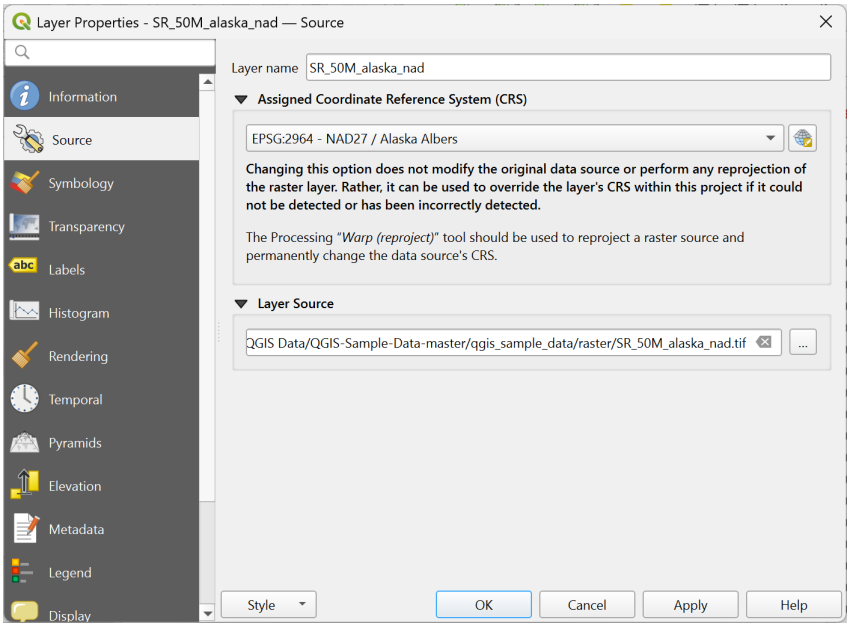
The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata for the current layer. Provided information are:

- general such as name in the project, source path, list of auxiliary files, last save time and size, the used provider
- custom properties, used to store in the active project additional information about the layer. Default custom properties include *Identify/format*, which influences how the results from using the *Identify features* tool over a raster layer are formatted. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- based on the provider of the layer: extent, width and height, data type, GDAL driver, bands statistics
- the Coordinate Reference System: name, units, method, accuracy, reference (i.e. whether it's static or dynamic)
- read from layer properties: data type, extent, width/height, compression, pixel size, statistics on bands, number of columns, rows and no-data values of the raster…
- picked from the *filled metadata*: access, extents, links, contacts, history…

13.1.2 Source Properties

The *Source* tab displays basic information about the selected raster, including:

- the *Layer name* to display in the *Layers Panel*;
- the *Coordinate Reference System*: Displays the layer's *Coordinate Reference System (CRS)*. You can change the layer's CRS, by selecting a recently used one in the drop-down list or clicking on the Select CRS button (see *Koordinačių atskaitos sistemos parinkiklis*). Use this process only if the layer CRS is a wrong or not specified. If you wish to reproject your data, use a reprojection algorithm from Processing or *Save it as new dataset*.
- Depending on the data provider, a *Layer source* group indicates the path to the source of the dataset and allows for replacing the loaded layer:
  - When the layer is stored as file on disk, edit the path shown in the text box or press … Browse to select another file on the disk
  - When the layer is provided by an ArcGIS MapServer service, it is possible to modify its *authentication settings*, while keeping unchanged the details for connecting to the service.
  - For an XYZ remote layer, it is possible to modify at the layer level any of its *connection details* (URL, authentication settings, zoom levels, resolution, …). Modifications are applied to the layer without altering the original connection settings.



13.1 Fig. : Raster Layer Properties - Source Dialog

13.1.3 Symbology Properties

The raster layer symbology tab is made of three different sections:

- The *Band rendering* where you can control the renderer type to use
- The *Layer rendering* to apply effects on rendered data
- The *Resampling* methods to optimize rendering on map

Band rendering

QGIS offers many different *Render types*. The choice of renderer depends on the data type and the information you'd like to highlight.

1. *Multiband color* - if the file comes with several bands (e.g. a satellite image with several bands).
2. *Paletted/Unique values* - for single band files that come with an indexed palette (e.g. a digital topographic map) or for general use of palettes for rendering raster layers.
3. *Singleband gray* - (one band of) the image will be rendered as gray. QGIS will choose this renderer if the file is neither multiband nor paletted (e.g. a shaded relief map).
4. *Singleband pseudocolor* - this renderer can be used for files with a continuous palette or color map (e.g. an elevation map).
5. *Single color* - the raster layer will be rendered with a single color.
6. *Hillshade* - Creates hillshade from a band.
7. *Contours* - Generates contours on the fly for a source raster band.

Multiband color

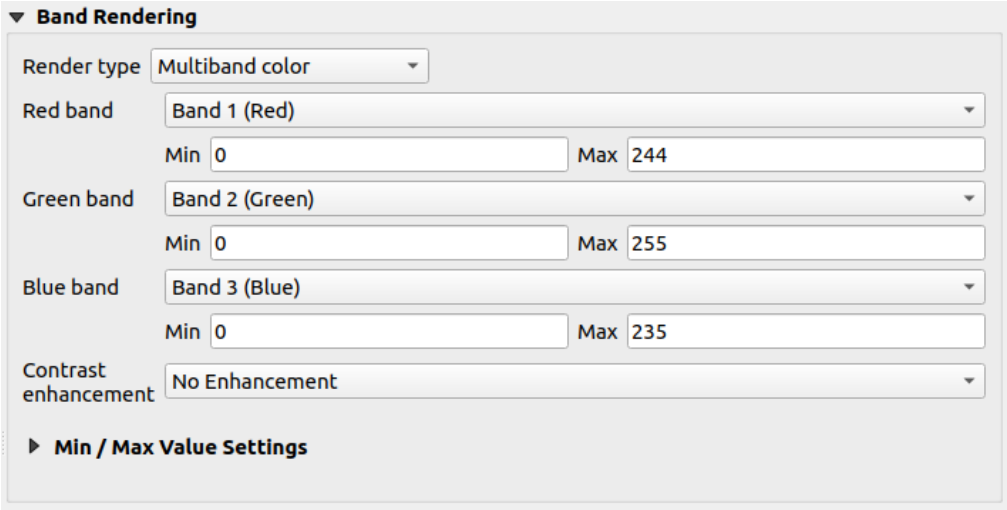
With the multiband color renderer, three selected bands from the image will be used as the red, green or blue component of the color image. QGIS automatically fetches *Min* and *Max* values for each band of the raster and scales the coloring accordingly. You can control the value ranges in the *Min/Max Value Settings* section.

A *Contrast enhancement* method can be applied to the values: ‚No enhancement', ‚Stretch to MinMax', ‚Stretch and clip to MinMax' and ‚Clip to min max'.

Pastaba

Contrast enhancement

When adding GRASS rasters, the option *Contrast enhancement* will always be set automatically to *stretch to min max*, even if this is set to another value in the QGIS general options.



13.2 Fig. : Raster Symbology - Multiband color rendering

Patarimas

Viewing a Single Band of a Multiband Raster

If you want to view a single band of a multiband image (for example, Red), you might think you would set the Green and Blue bands to *Not Set*. But the preferred way of doing this is to set the image type to *Singleband gray*, and then select Red as the *Gray band* to use.

Paletted/Unique values

This is the standard render option for singleband files that include a color table, where a certain color is assigned to each pixel value. In that case, the palette is rendered automatically.

It can be used for all kinds of raster bands, assigning a color to each unique raster value.

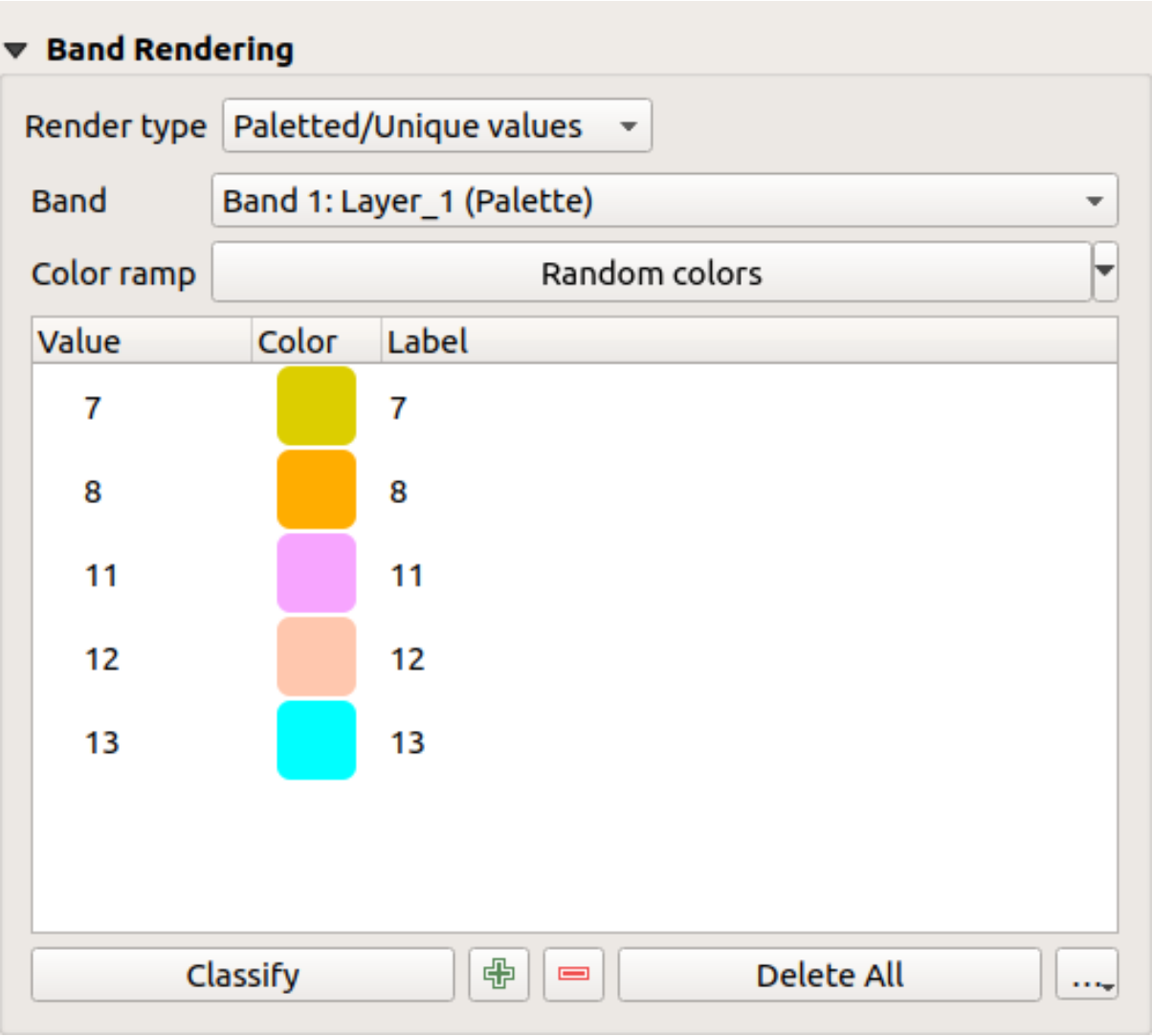
If you want to change a color, just double-click on the color and the *Select color* dialog appears.

It is also possible to assign labels to the colors. The label will then appear in the legend of the raster layer.

Right-clicking over selected rows in the color table shows a contextual menu to:

- *Change Color…* for the selection

- *Change Opacity…* for the selection
- *Change Label…* for the selection



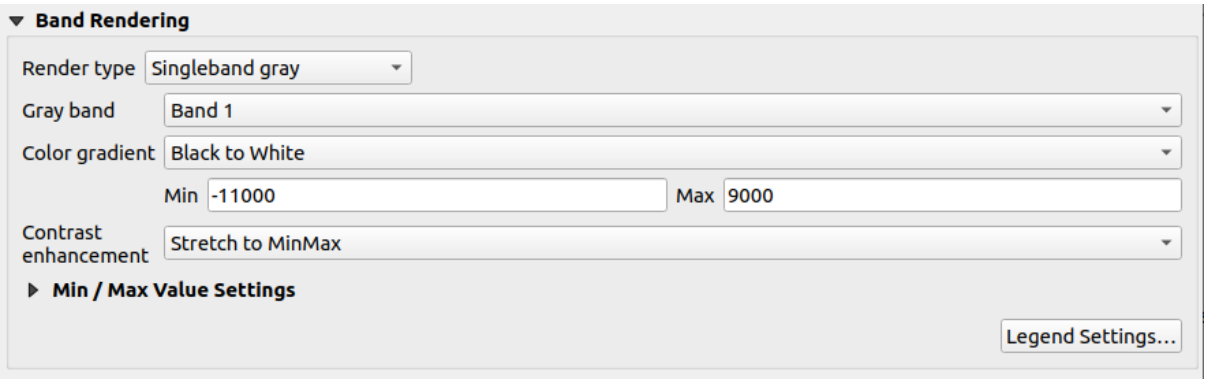
13.3 Fig. : Raster Symbology - Paletted unique value rendering

The pulldown menu, that opens when clicking the … (Advanced options) button below the color map to the right, offers color map loading (*Load Color Map from File…*) and exporting (*Export Color Map to File…*), and loading of classes (*Load Classes from Layer*).

Singleband gray

This renderer allows you to render a layer using only one band with a *Color gradient*: ‚Black to white‘ or ‚White to black‘. You can change the range of values to color (*Min* and *Max*) in the *Min/Max Value Settings*.

A *Contrast enhancement* method can be applied to the values: ‚No enhancement‘, ‚Stretch to MinMax‘, ‚Stretch and clip to MinMax‘ and ‚Clip to min max‘.

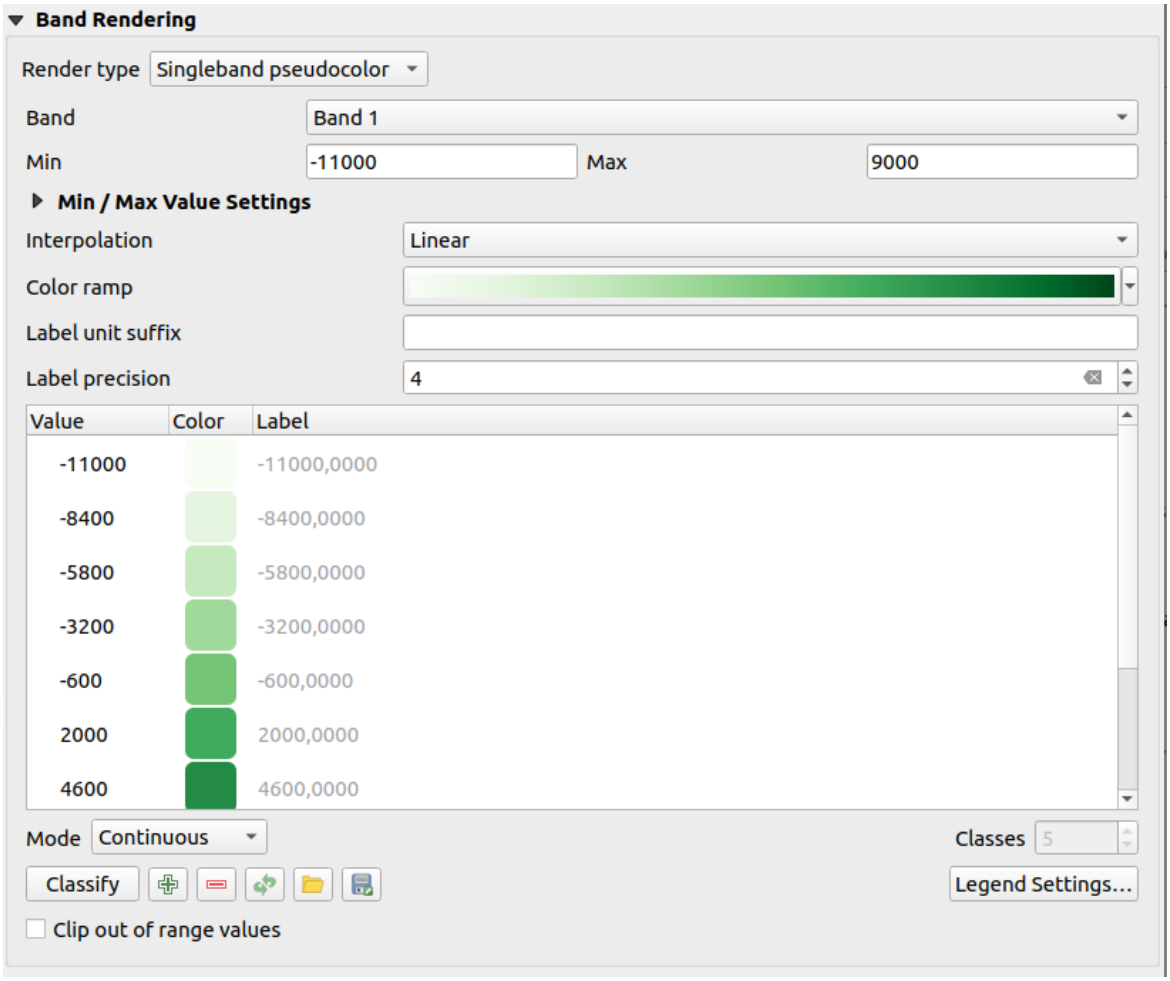


13.4 Fig. : Raster Symbology - Singleband gray rendering

Pixels are assigned a color based on the selected color gradient and the layer's legend (in the *Layers* panel and the layout *legend item*) is displayed using a continuous color ramp. Press *Legend settings...* if you wish to tweak the settings. More details at *Customize raster legend*.

Singleband pseudocolor

This is a render option for single-band files that include a continuous palette. You can also create color maps for a band of a multiband raster.



13.5 Fig. : Raster Symbology - Singleband pseudocolor rendering

Using a *Band* of the layer and a *values range*, you can now interpolate and assign representation color to pixels within classes. More at *Color ramp shader classification*.

Pixels are assigned a color based on the selected color ramp and the layer's legend (in the *Layers* panel and the layout *legend item*) is displayed using a continuous color ramp. Press *Legend settings…* if you wish to tweak the settings or instead use a legend with separated classes (and colors). More details at *Customize raster legend*.

Single color

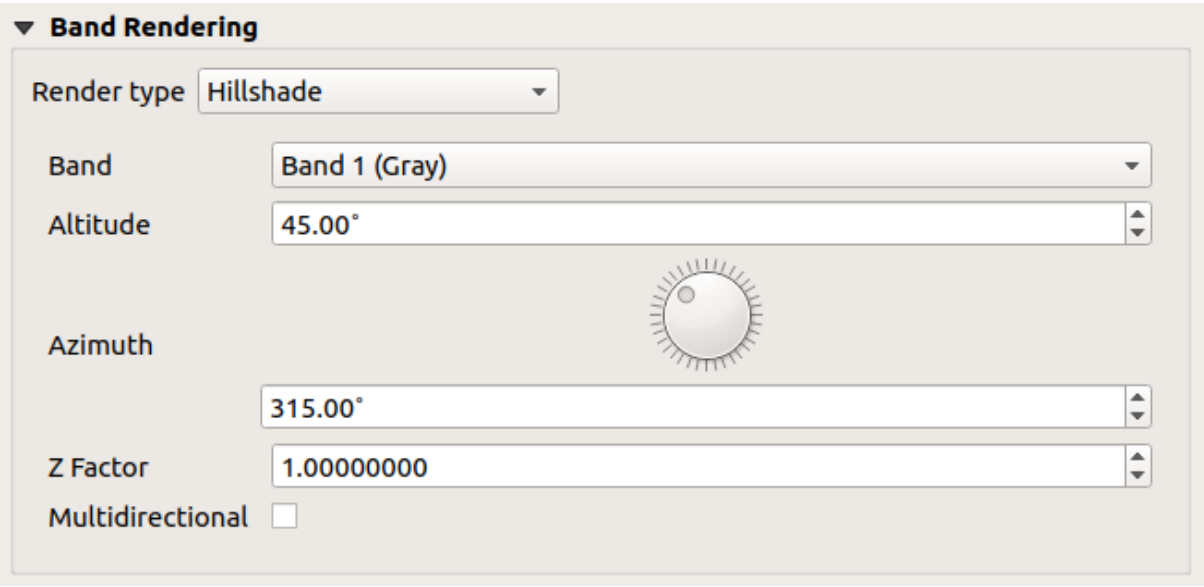
This renderer allows you to render a raster layer using *Single color*. This type of renderer is useful when you want to display a raster layer uniformly, without any variation in color based on pixel values.

The single color renderer can be used with both single-band and multiband raster layers. When used with multiband rasters, you can select which band to apply the single color to, effectively displaying that specific band uniformly across the entire layer.

13.6 Fig. : Raster Symbology - Single color rendering

Hillshade

Render a band of the raster layer using hillshading.



13.7 Fig. : Raster Symbology - Hillshade rendering

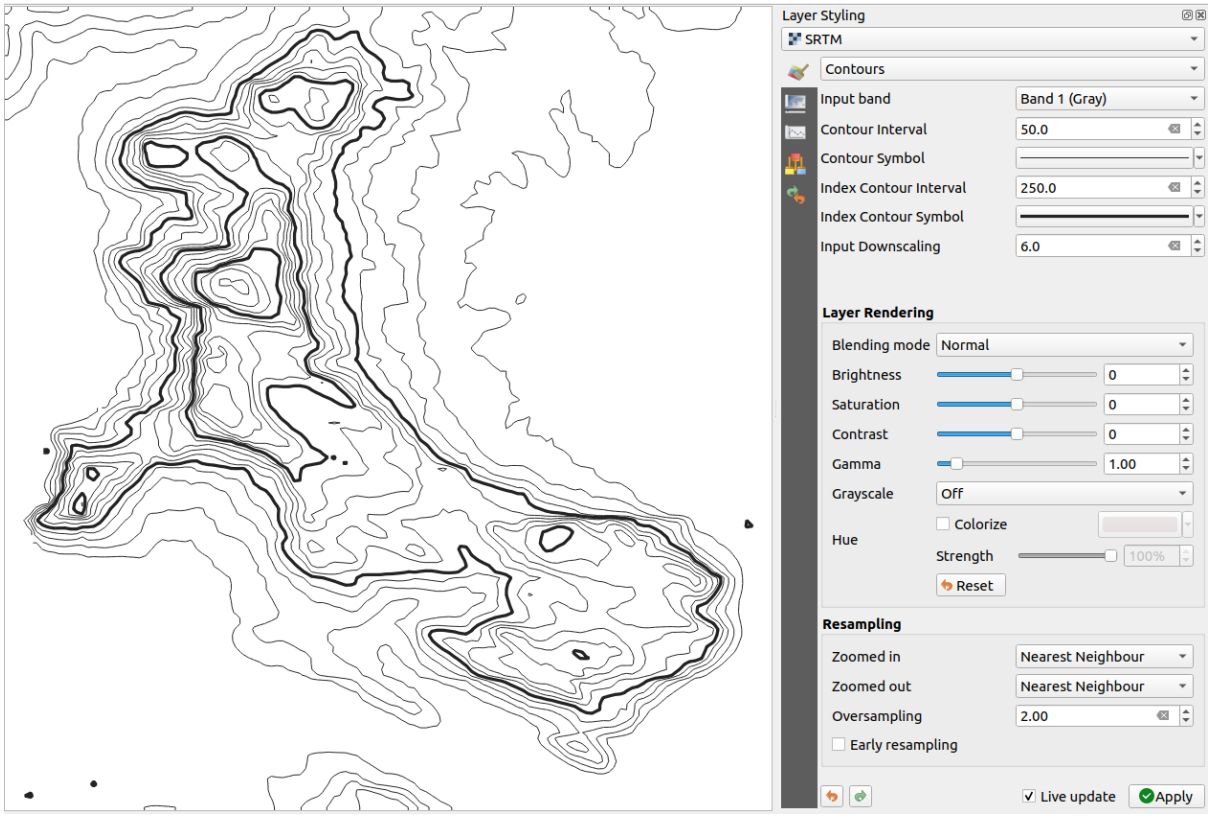
Parinktys:

- *Band*: The raster band to use.
- *Altitude*: The elevation angle of the light source (default is `45°`).

- *Azimuth*: The azimuth of the light source (default is `315°`).
- *Z Factor*: Scaling factor for the values of the raster band (default is `1`).
- *Multidirectional*: Specify if multidirectional hillshading is to be used (default is `off`).

Contours

This renderer draws contour lines that are calculated on the fly from the source raster band.



13.8 Fig. : Raster Symbology - Contours rendering

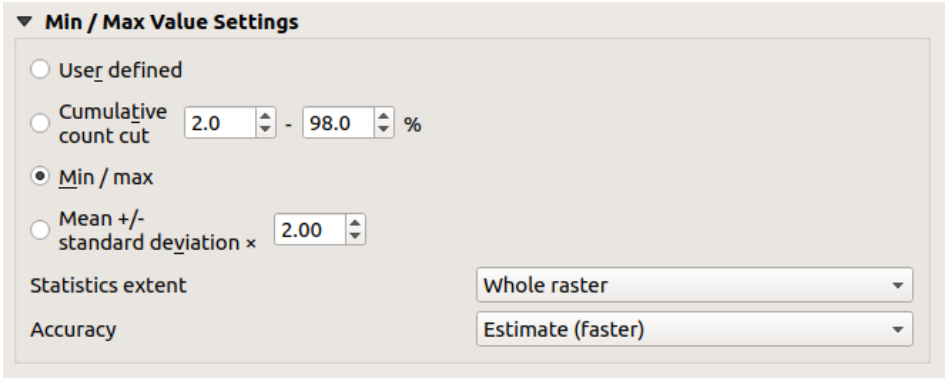
Parinktys:

- *Input band*: the raster band to use.
- *Contour interval*: the distance between two consecutive contour lines
- *Contour symbol*: the *symbol* to apply to the common contour lines.
- *Index contour interval*: the distance between two consecutive **index contours**, that is the lines shown in a distinctive manner for ease of identification, being commonly printed more heavily than other contour lines and generally labeled with a value along its course.
- *Index contour symbol*: the symbol to apply to the index contour lines
- *Input downscaling*: Indicates by how much the renderer will scale down the request to the data provider (default is `4.0`).

  For example, if you generate contour lines on input raster block with the same size as the output raster block, the generated lines would contain too much detail. This detail can be reduced by the „downscale“ factor, requesting lower resolution of the source raster. For a raster block 1000x500 with downscale 10, the renderer will request raster 100x50 from provider. Higher downscale makes contour lines more simplified (at the expense of losing some detail).

Setting the min and max values

By default, QGIS reports the *Min* and *Max* values of the band(s) of the raster. A few very low and/or high values can have a negative impact on the rendering of the raster. The *Min/Max Value Settings* frame helps you control the rendering.



13.9 Fig. : Raster Symbology - Min and Max Value Settings

Galimos parinktys:

- *User defined*: The default *Min* and *Max* values of the band(s) can be overridden
- *Cumulative count cut*: Removes outliers. The standard range of values is `2%` to `98%`, but it can be adapted manually.
- *Min / max*: Uses the whole range of values in the image band.
- *Mean +/- standard deviation x*: Creates a color table that only considers values within the standard deviation or within multiple standard deviations. This is useful when you have one or two cells with abnormally high values in a raster layer that impact the rendering of the raster negatively.

Calculations of the min and max values of the bands are made based on the:

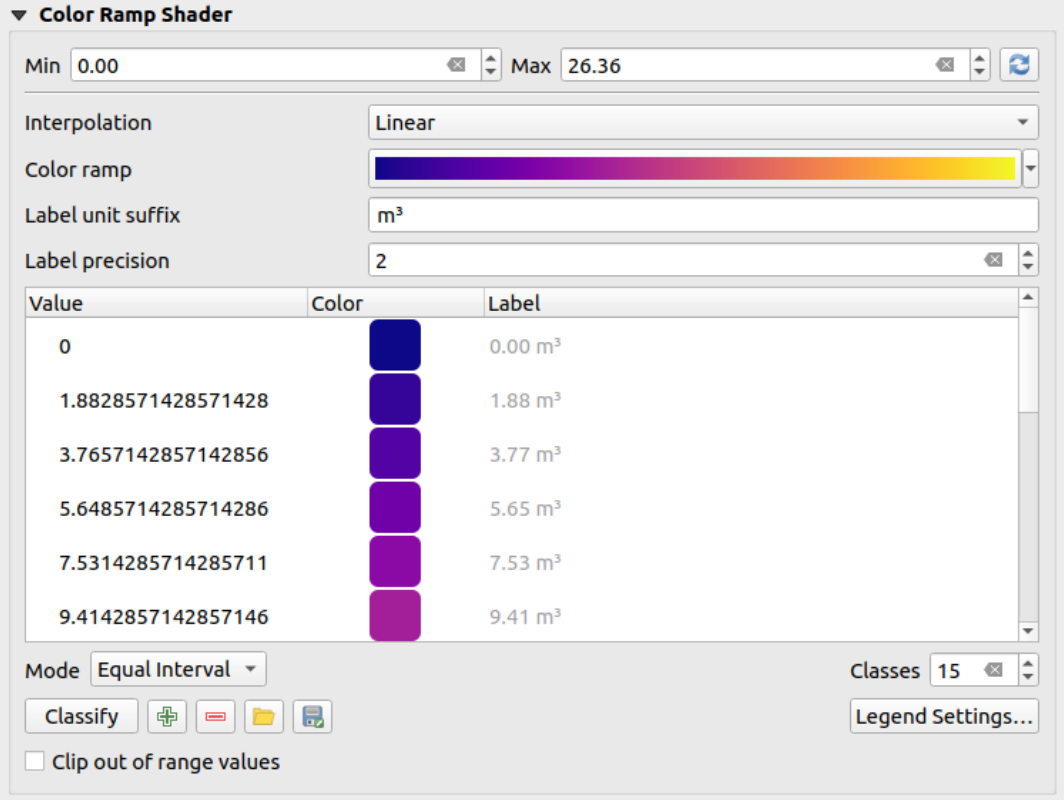
- *Statistics extent*: it can be *Whole raster*, *Current canvas* or *Updated canvas*. *Updated canvas* means that min/max values used for the rendering will change with the canvas extent (dynamic stretching).
- *Accuracy*, which can be either *Estimate (faster)* or *Actual (slower)*.

Pastaba

For some settings, you may need to press the *Apply* button of the layer properties dialog in order to display the actual min and max values in the widgets.

Color ramp shader classification

This method can be used to classify and represent scalar dataset (raster or mesh contour) based on their values. Given a *color ramp* and a number of classes, it generates intermediate color map entries for class limits. Each color is mapped with a value interpolated from a range of values and according to a classification mode. The scalar dataset elements are then assigned their color based on their class.



13.10 Fig. : Classifying a dataset with a color ramp shader

1. A *Min* and *Max* values must be defined and used to interpolate classes bounds. By default QGIS detects them from the dataset but they can be modified.
2. The *Interpolation* entry defines how scalar elements are assigned their color :
   - *Discrete* (a <= symbol appears in the header of the *Value* column): The color is taken from the closest color map entry with equal or higher value
   - *Linear*: The color is linearly interpolated from the color map entries above and below the pixel value, meaning that to each dataset value corresponds a unique color
   - *Exact* (a = symbol appears in the header of the *Value* column): Only pixels with value equal to a color map entry are applied a color; others are not rendered.
3. The *Color ramp* widget helps you select the color ramp to assign to the dataset. As usual with *this widget*, you can create a new one and edit or save the currently selected one. The name of the color ramp will be saved in the configuration.
4. The *Label unit suffix* adds a label after the value in the legend, and the *Label precision* controls the number of decimals to display.
5. The classification *Mode* helps you define how values are distributed across the classes:
   - *Equal interval*: Provided the *Number of classes*, limits values are defined so that the classes all have the same magnitude.
   - *Continuous*: Classes number and color are fetched from the color ramp stops; limits values are set following stops distribution in the color ramp.
   - *Quantile*: Provided the *Number of classes*, limits values are defined so that the classes have the same number of elements. Not available with *mesh layers*.
6. You can then *Classify* or tweak the classes:
   - The button Add values manually adds a value to the table.

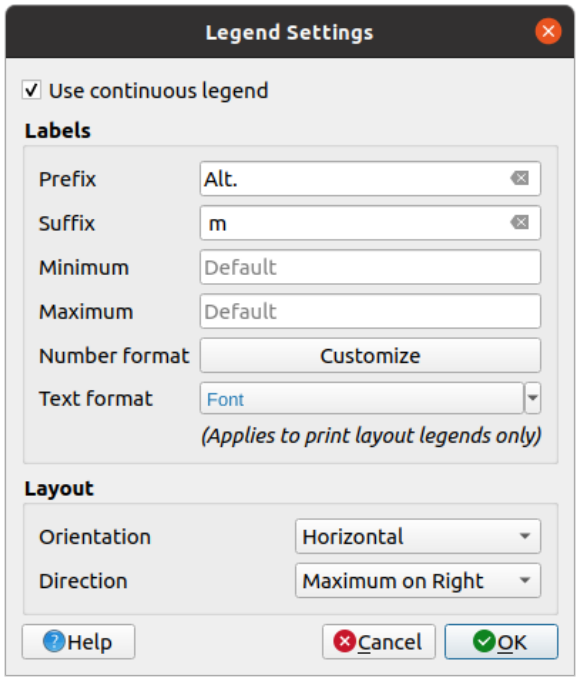
- The button Remove selected row deletes selected values from the table.
- Double clicking in the *Value* column lets you modify the class value.
- Double clicking in the *Color* column opens the dialog *Change color*, where you can select a color to apply for that value.
- Double clicking in the *Label* column to modify the label of the class, but this value won't be displayed when you use the identify feature tool.
- Right-clicking over selected rows in the color table shows a contextual menu to *Change Color…* and *Change Opacity…* for the selection.

You can use the buttons Load color map from file or Export color map to file to load an existing color table or to save the color table for later use.

7. With linear *Interpolation*, you can also configure:
   - *Clip out of range values*: By default, the linear method assigns the first class (respectively the last class) color to values in the dataset that are lower than the set *Min* (respectively greater than the set *Max*) value. Check this setting if you do not want to render those values.
   - *Legend settings*, for display in the *Layers* panel and the layout *legend item*. More details at *Customize raster legend*.

Customize raster legend

When applying a color ramp to a raster or a mesh layer, you may want to display a legend showing the classification. By default, QGIS displays a continuous color ramp with min and max values in the *Layers* panel and the layout *legend item*. This can be customized using the *Legend settings* button in the classification widget.



13.11 Fig. : Modifying a raster legend

In this dialog, you can set whether to *Use continuous legend*: if unchecked, the legend displays separated colors corresponding to the different classes applied. This option is not available for raster *singleband gray* symbology.

Checking the *Use continuous legend* allows you to configure both the labels and layout properties of the legend.

Labels

- Add a *Prefix* and a *Suffix* to the labels
- Modify the *Minimum* and a *Maximum* values to show in the legend
- *Customize* the *Number format*
- *Customize* the *Text format* to use in the print layout legend.

Layout

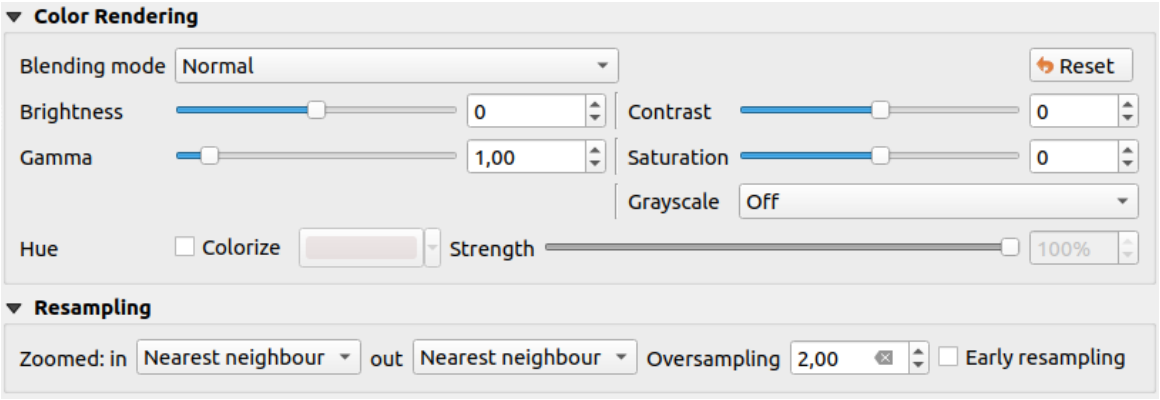
- Control the *Orientation* of the legend color ramp; it can be **Vertical** or **Horizontal**
- Control the *Direction* of the values depending on the orientation:
  - If vertical, you can display the **Maximum on top** or the **Minimum on top**
  - If horizontal, you can display the **Maximum on right** or the **Minimum on right**

Layer rendering

Over the symbology type applied to the layer band(s), you can achieve special rendering effects for the whole raster file(s):

- Use one of the blending modes (see *Suliejimo režimai*)
- Set custom *Brightness*, *Saturation*, *Gamma* and *Contrast* to colors.
- With the *Invert colors*, the layer is rendered with opposite colors. Handy, for example, to switch out-of-the box OpenStreetMap tiles to dark mode.
- Turn the layer to *Grayscale* option either ‚By lightness', ‚By luminosity' or ‚By average'.
- *Colorize* and adjust the *Strength* of *Hue* in the color table

Press *Reset* to remove any custom changes to the layer rendering.



13.12 Fig. : Raster Symbology - Layer rendering and Resampling settings

Resampling

The *Resampling* option has effect when you zoom in and out of an image. Resampling modes can optimize the appearance of the map. They calculate a new gray value matrix through a geometric transformation.

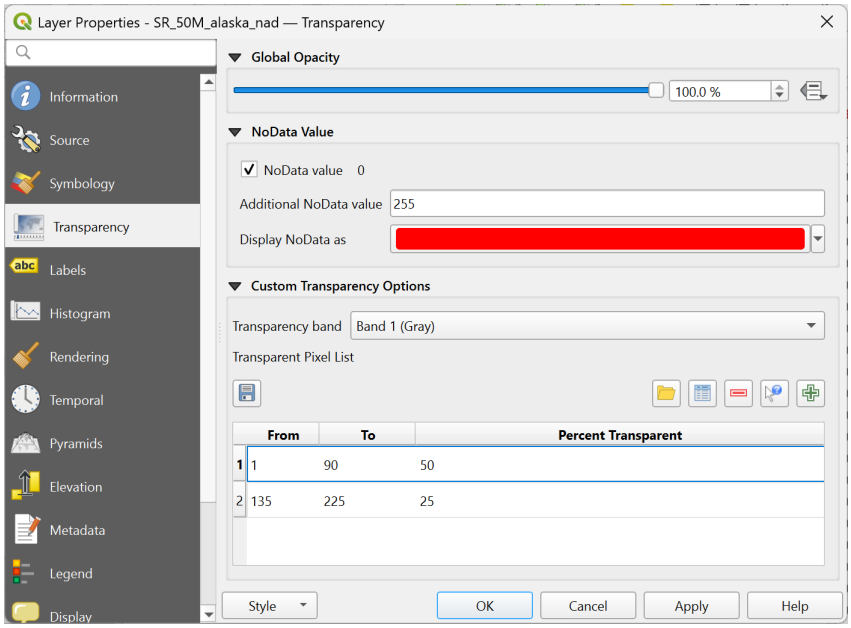
When applying the ‚Nearest neighbour' method, the map can get a pixelated structure when zooming in. This appearance can be improved by using the ‚Bilinear (2x2 kernel)' or ‚Cubic (4x4 kernel)' method, which cause sharp edges to be blurred. The effect is a smoother image. This method can be applied to for instance digital topographic raster maps.

Early resampling: allows to calculate the raster rendering at the provider level where the resolution of the source is known, and ensures a better zoom in rendering with QGIS custom styling. Really convenient for tile rasters loaded using an *interpretation method*.

13.1.4 Transparency Properties

QGIS provides capabilities to set the *Transparency* level of a raster layer.

Use the *Global opacity* slider to set to what extent the underlying layers (if any) should be visible through the current raster layer. This is very useful if you overlay raster layers (e.g., a shaded relief map overlayed by a classified raster map). This will make the look of the map more three dimensional. The opacity of the raster can be data-defined, and vary e.g. depending on the visibility of another layer, by temporal variables, on different pages of an atlas, …



13.13 Fig. : Raster Transparency

With *No data value* QGIS reports the original source no data value (if defined) which you can consider as is in the rendering. Additionally, you can enter a raster value that should be treated as an *Additional no data value*. The *Display no data as* color selector allows you to apply a custom color to no data pixels, instead of the default transparent rendering.

An even more flexible way to customize the transparency is available in the *Custom transparency options* section:

- Use *Transparency band* to apply transparency for an entire band.
- Provide a list of pixels to make transparent with corresponding levels of transparency:

  1. Click the Add values manually button. A new row will appear in the pixel list.

2. For single-band based symbology (e.g. DEMs), enter the **From** and **To** values and adjust the **Percent Transparent** to apply.
3. For multiband based symbology (e.g. RGB images) enter the **Red**, **Green** and **Blue** values of the pixel and adjust the **Percent Transparent** to apply. QGIS supports **Tolerance** for pixel values, when defining transparency. This means that pixels with color close to the specified RGB values can also be made transparent. Note that this feature applies only to multiband rasters.
4. Alternatively, you can fetch the pixel values directly from the raster using the Add values from display button. Then enter the transparency value.
5. Repeat the steps to adjust more values with custom transparency.
6. Press the *Apply* button and have a look at the map.

As you can see, it is quite easy to set custom transparency, but it can be quite a lot of work. Therefore, you can use the button Export to file to save your transparency list to a file. The button Import from file loads your transparency settings and applies them to the current raster layer.

13.1.5 Labels Properties

The *Labels* properties provides you with all the needed and appropriate capabilities to configure smart labeling on raster layers. This dialog can also be accessed from the *Layer Styling* panel.

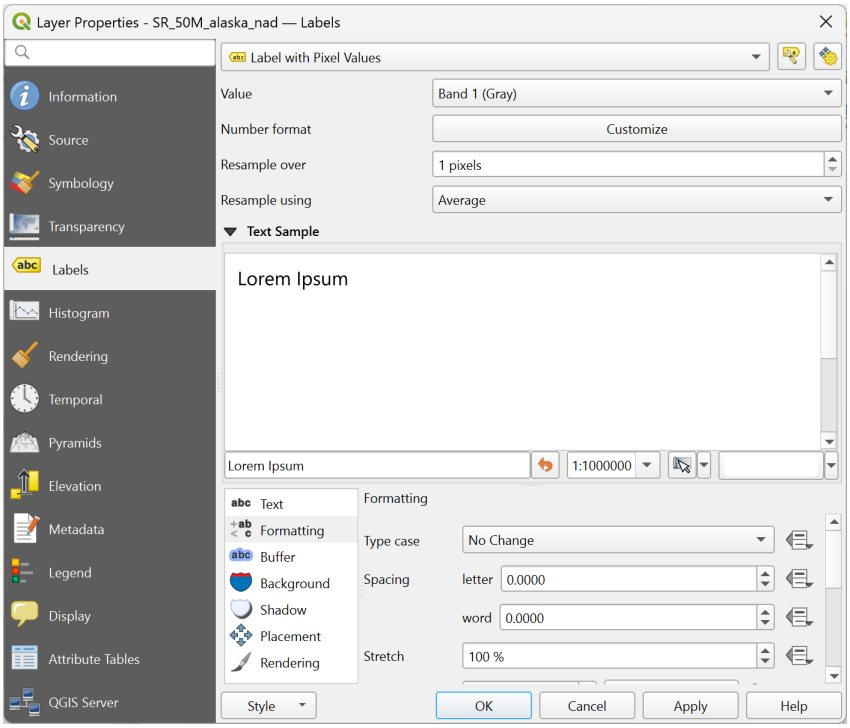
At the top of the dialog, you have:

- a combobox for selecting the appropriate labeling method for the active layer
- the Configure project labeling rules button: helps you control interactions between labels and features across the layers in the project. More details at *Configuring project labeling rules*.
- the Automated placement settings (applies to all layers) button: configure general properties on label placement and conflicts resolution. More details at *Setting the automated placement engine*.

The first step is to choose the labeling method from the drop-down list. Available methods are:

- *No labels*: the default value, showing no labels from the layer
- *Label with pixel values*: Show labels on the map using a band.

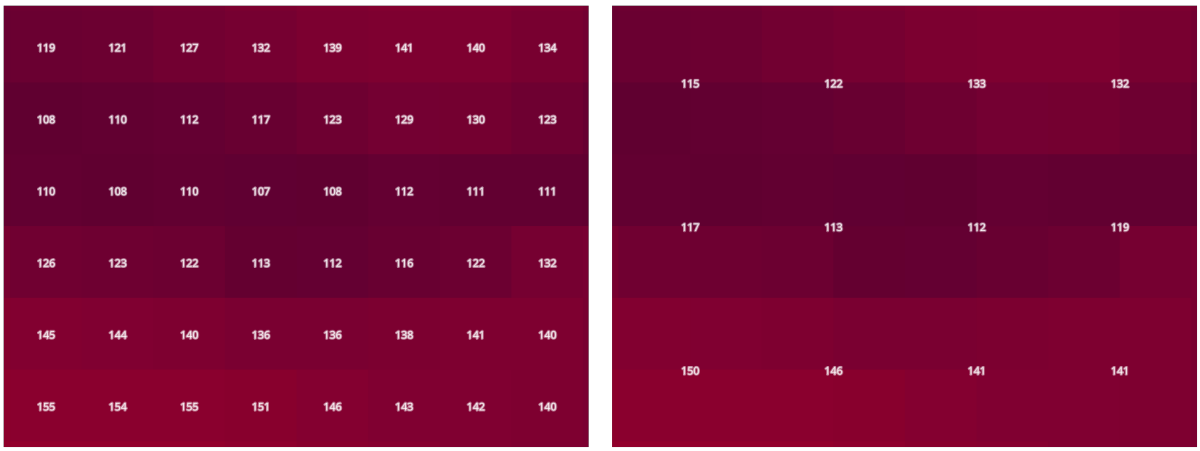
Using the *Label with pixel values* option, the following dialog opens.



13.14 Fig. : Raster layer labeling settings

At the top of the dialog:

- A *Value* drop-down list allows you to select the band to take the values from
- Press *Customize* to configure a *proper number formatting*
- By default, the displayed value represents individual pixel band value. With *Resample over*, you can compute the value from the neighbouring pixels (setting `2` means `2*2=4` pixels) using a statistical method set in the *Resample using* widget.



13.15 Fig. : Pixels labeled using various resampling options

Below are displayed options to customize the labels, under various tabs:

- abc *Text*
- +ab < c *Formatting*
- abc *Buffer*

- *Background*
- *Shadow*
- *Placement*
- *Rendering*

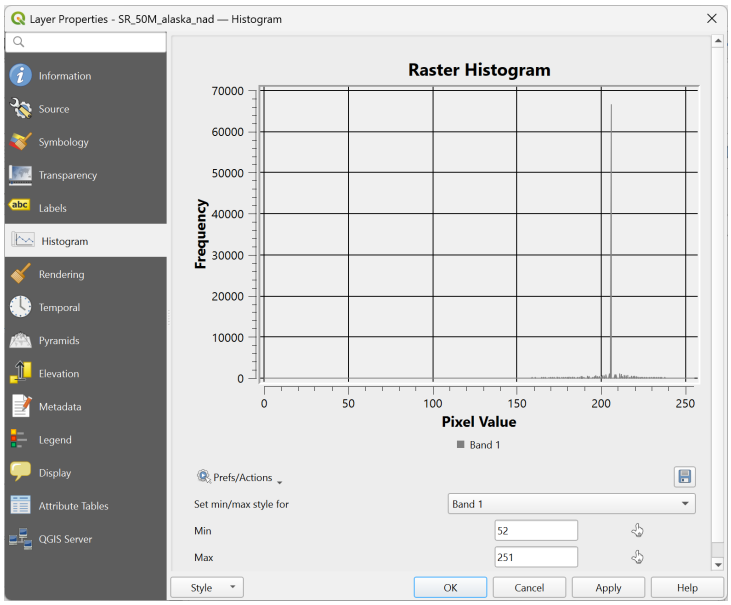
Description of how to set each property is exposed at *Užrašo nustatymas*.

13.1.6 Histogram Properties

The *Histogram* tab allows you to view the distribution of the values in your raster. The histogram is generated when you press the *Compute Histogram* button. All existing bands will be displayed together. You can save the histogram as an image with the button.

At the bottom of the histogram, you can select a raster band in the drop-down menu and *Set min/max style for* it. The *Prefs/Actions* drop-down menu gives you advanced options to customize the histogram:

- With the *Visibility* option, you can display histograms for individual bands. You will need to select the option *Show selected band*.
- The *Min/max options* allow you to ‚Always show min/max markers', to ‚Zoom to min/max' and to ‚Update style to min/max'.
- The *Actions* option allows you to ‚Reset' or ‚Recompute histogram' after you have changed the min or max values of the band(s).



13.16 Fig. : Raster Histogram

13.1.7 Rendering Properties

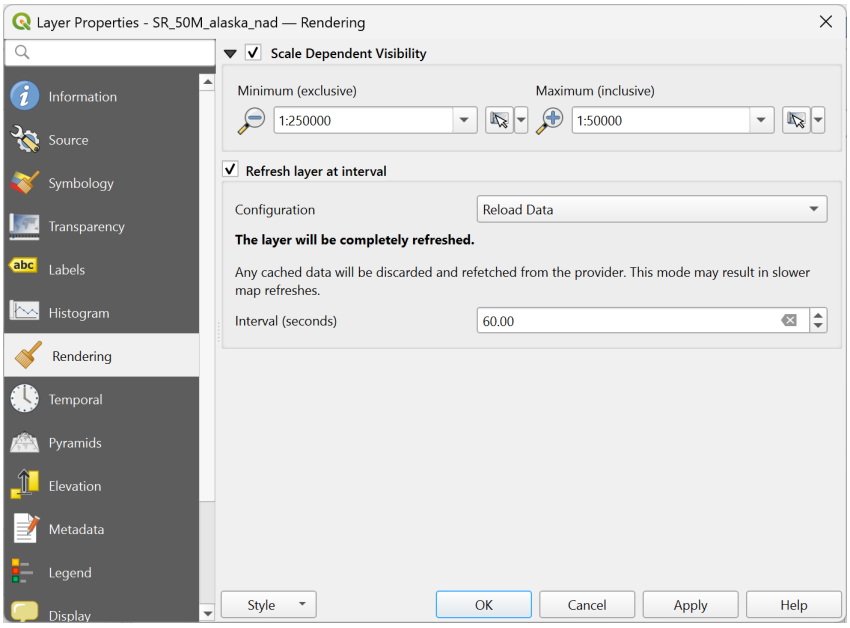
In the *Rendering* tab, it's possible to:

- set *Scale dependent visibility* for the layer: You can set the *Maximum (inclusive)* and *Minimum (exclusive)* scales, defining a range of scales in which the layer will be visible. It will be hidden outside this range. The Set to current canvas scale button helps you use the current map canvas scale as a boundary. See *Nuo mastelio priklausomo matomumo parinkimas* for more information.

  > **Pastaba**
  >
  > You can also activate scale dependent visibility on a layer from within the *Layers* panel: right-click on the layer and in the contextual menu, select *Set Layer Scale Visibility*.

- *Refresh layer at interval*: controls whether and how regular a layer can be refreshed. Available *Configuration* options are:
  - *Reload data*: the layer will be completely refreshed. Any cached data will be discarded and refetched from the provider. This mode may result in slower map refreshes.
  - *Redraw layer only*: this mode is useful for animation or when the layer's style will be updated at regular intervals. Canvas updates are deferred in order to avoid refreshing multiple times if more than one layer has an auto update interval set.

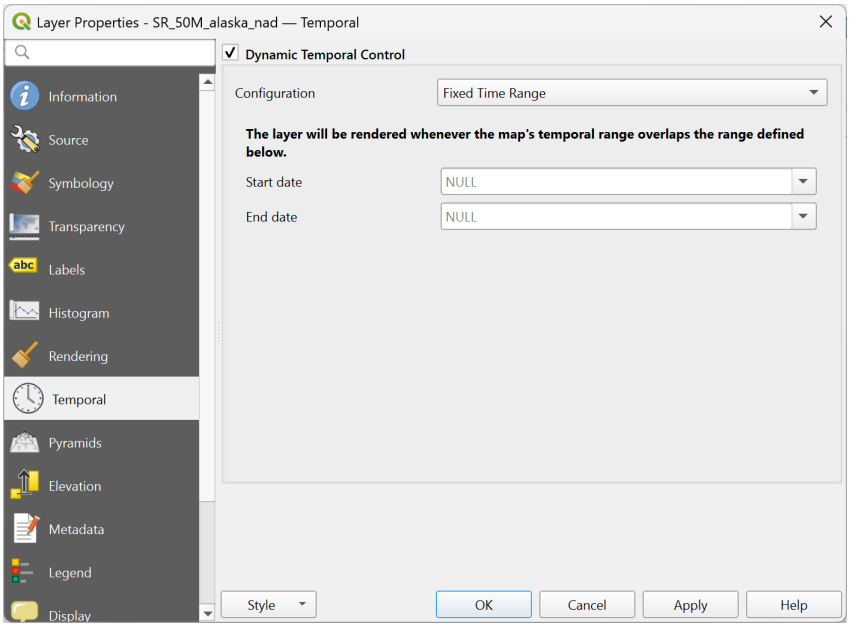
  It is also possible to set the *Interval (seconds)* between consecutive refreshments.



13.17 Fig. : Raster Rendering Properties

13.1.8 Laiko savybės

The *Temporal* tab provides options to control the rendering of the layer over time. Such dynamic rendering requires the *temporal navigation* to be enabled over the map canvas.



13.18 Fig. : Raster Temporal Properties

Check the *Dynamic Temporal Control* option and set whether the layer redraw should be:

- *Automatic*: the rendering is controlled by the underlying data provider if it suppports temporal data handling. E.g. this can be used with WMS-T layers or PostgreSQL rasters.
- *Fixed Date/Time*: only show the raster layer at a single, specific date/time. This avoids having to enter the same value for both the start and end of the temporal range.
- *Fixed time range*: only show the raster layer if the animation time is within a *Start date* and *End date* range.
- *Fixed Time Range Per Band*: only shows a band when the current animation time is between its *Begin* and *End* date range. This option allows you to either manually set these time ranges for each band or use the ε button to automatically generate datetime values, enabling detailed temporal analysis and visualization. This mode is particularly useful for working with raster layers where each band corresponds to a specific time period, such as NetCDF files.
- *Represents Temporal Values*: interprets each pixel in the raster layer as a datetime value. When this temporal mode is active, pixels that do not fall within the temporal range specified in the render context will be hidden, ensuring that only temporally relevant data is displayed. This mode is effective for:
  - Analyzing land use changes, like observing deforestation patterns.
  - Studying flooding by comparing water coverage across different times.
  - Evaluating movement costs in terrain analysis, for example, using GRASS GIS's r.walk tool to calculate travel costs across a landscape.
- *Redraw layer only*: the layer is redrawn at each new animation frame. It's useful when the layer uses time-based expression values for renderer settings (e.g. data-defined renderer opacity, to fade in/out a raster layer).

13.1.9 Pyramids Properties

High resolution raster layers can slow navigation in QGIS. By creating lower resolution copies of the data (pyramids), performance can be considerably improved, as QGIS selects the most suitable resolution to use depending on the zoom level.

You must have write access in the directory where the original data is stored to build pyramids.

From the *Resolutions* list, select resolutions at which you want to create pyramid levels by clicking on them.

If you choose **Internal (if possible)** from the *Overview format* drop-down menu, QGIS tries to build pyramids internally.

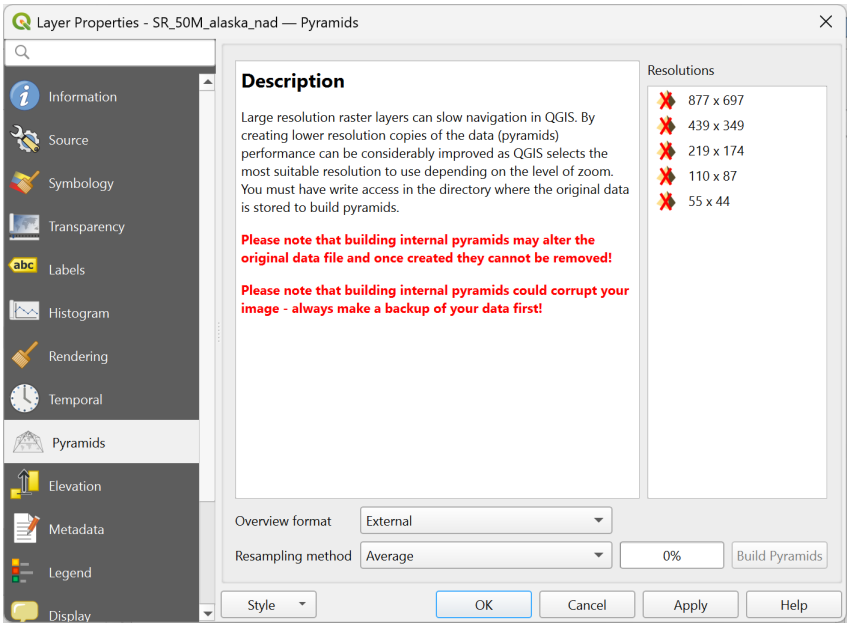
> **Pastaba**
>
> Please note that building pyramids may alter the original data file, and once created they cannot be removed. If you wish to preserve a ‚non-pyramided' version of your raster, make a backup copy prior to pyramid building.

If you choose **External** and **External (Erdas Imagine)** the pyramids will be created in a file next to the original raster with the same name and a `.ovr` extension.

Several *Resampling methods* can be used for pyramid calculation:

- Nearest Neighbour
- Average
- Gauss
- Cubic
- Cubic Spline
- Laczos
- Mode
- None

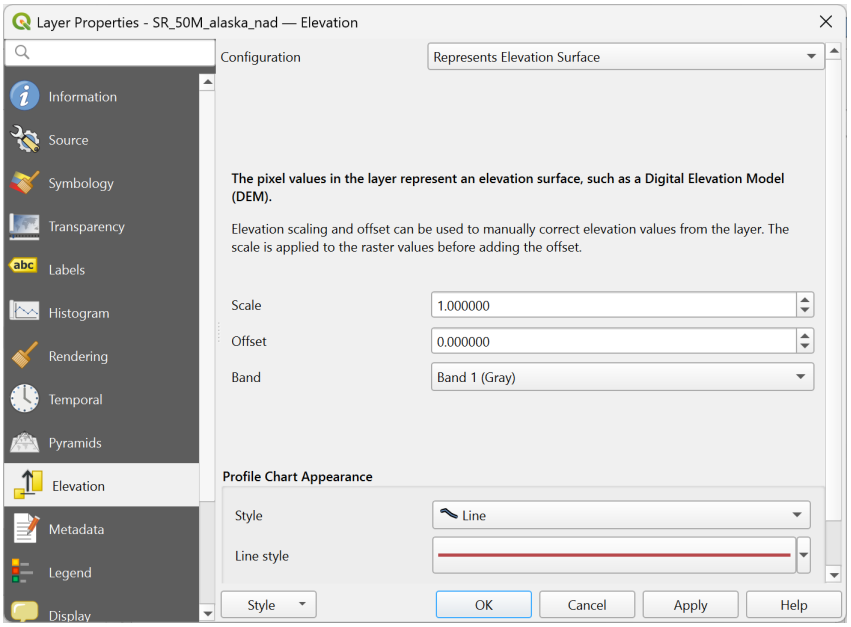
Finally, click *Build Pyramids* to start the process.



13.19 Fig. : Raster Pyramids

13.1.10 Aukščio savybės

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view* and its appearance in the *profile tool charts*. Specifically, you can choose to *Disable* this configuration if the layer does not contain elevation data or you can set:



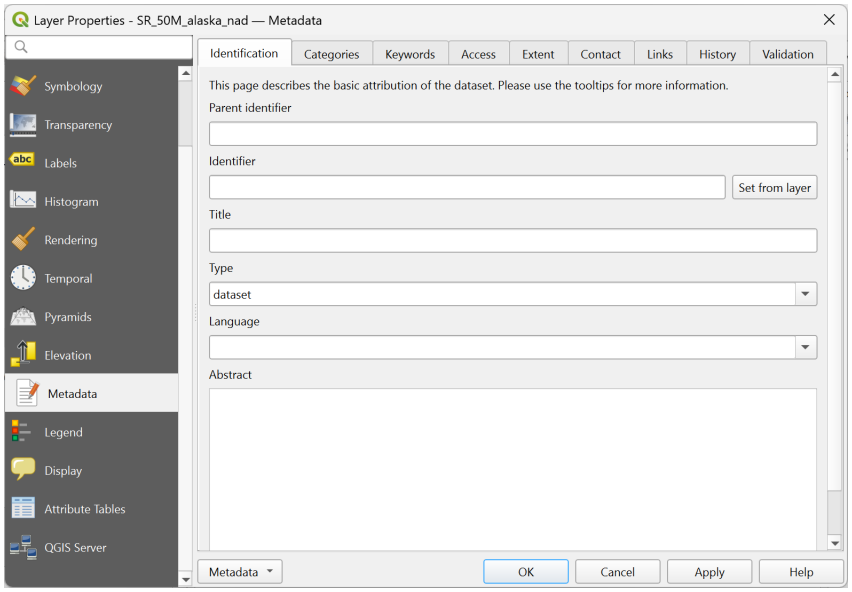
13.20 Fig. : Raster Elevation Properties

- *Represents Elevation Surface*: whether the raster layer represents a height surface (e.g DEM) and the pixel values should be interpreted as elevations. Choose this option if you want to display a raster in an *elevation profile view*. You will also need to fill in the *Band* to pick values from and can apply a *Scale* factor and an *Offset*.
- *Fixed Elevation Range*: The raster layer (or selected raster band) is associated with a fixed elevation range. This mode can be used when a layer has a single fixed elevation or a range (slice) of elevation values. If a range is specified, pixels will be extruded over this range. You can set the *Lower* and *Upper* elevation range values for the layer, and specify whether the lower or upper *Limits* are inclusive or exclusive.
- *Fixed Elevation Range Per Band*: Each band in the raster can have a fixed elevation range associated with it. This is designed for data sources that expose elevation-related data in bands, such as NetCDF files. For example, a raster with temperature data at different ocean depths. When rendering, the uppermost matching band will be selected and used for the layer's data. This feature is exposed as a user-editable table for raster bands with lower and upper values. Users can either populate the lower and upper values manually or use an *Expression* to auto-fill all band values based on expression. The expression-based fill allows you to design expressions that extract useful information from band names. For example, extracting the depth value from a band name like „Band 001: depth=-5500 (meters)".
- *Dynamic Elevation Range Per Band*: This mode calculates elevation ranges for raster bands dynamically using QGIS expressions. It's ideal for datasets where elevation values follow a consistent pattern across bands (like equally spaced vertical layers), eliminating the need to manually assign fixed elevations. Instead of entering individual values for each band, you define expressions for the *Lower* and *Upper* elevation bounds. These expressions can use variables like `@band`, `@band_name`, or `@band_description` automatically computing the elevation range based on each band's properties.
- *Profile Chart Appearance*: controls the rendering of the raster elevation data in the profile chart. The profile *Style* can be set as:
  - a *Line* with a specific *Line style*

- an elevation surface rendered using a fill symbol either above (*Fill above*) or below (*Fill below*) the elevation curve line. The surface symbology is represented using:
  - a *Fill style*
  - and a *Limit*: the maximum (respectively minimum) altitude determining how high the fill surface will be

13.1.11 Metaduomenų savybės

The *Metadata* tab provides you with options to create and edit a metadata report on your layer. See *Metaduomenys* for more information.



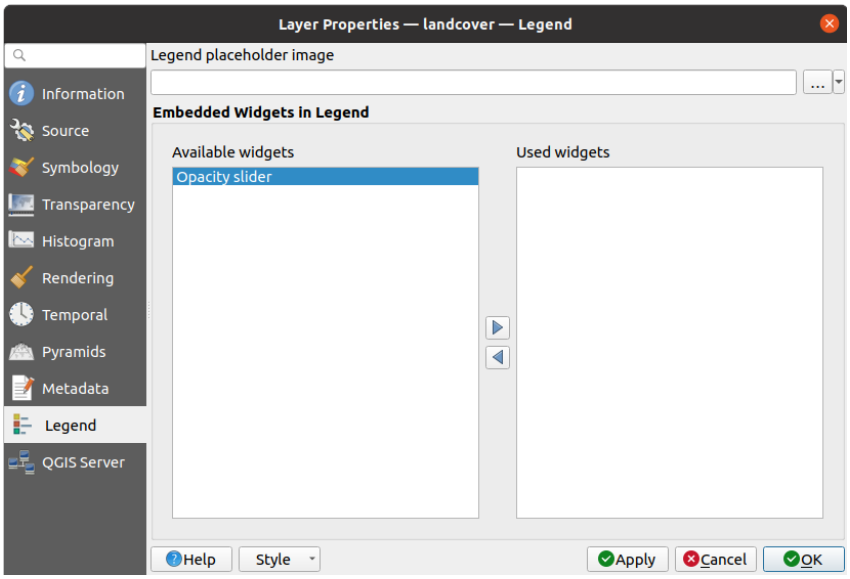
13.21 Fig. : Raster Metadata

13.1.12 Legend Properties

The *Legend* tab provides you with advanced settings for the *Layers panel* and/or the *print layout legend*. These options include:

- Depending on the symbology applied to the layer, you may end up with several entries in the legend, not necessarily readable/useful to display. The *Legend placeholder image* helps you *select an image* for replacement, displayed both in the *Layers* panel and the print layout legend.
- The *Embedded widgets in Legend* provides you with a list of widgets you can embed within the layer tree in the Layers panel. The idea is to have a way to quickly access some actions that are often used with the layer (setup transparency, filtering, selection, style or other stuff…).

  By default, QGIS provides a transparency widget but this can be extended by plugins that register their own widgets and assign custom actions to layers they manage.



13.22 Fig. : Raster Legend

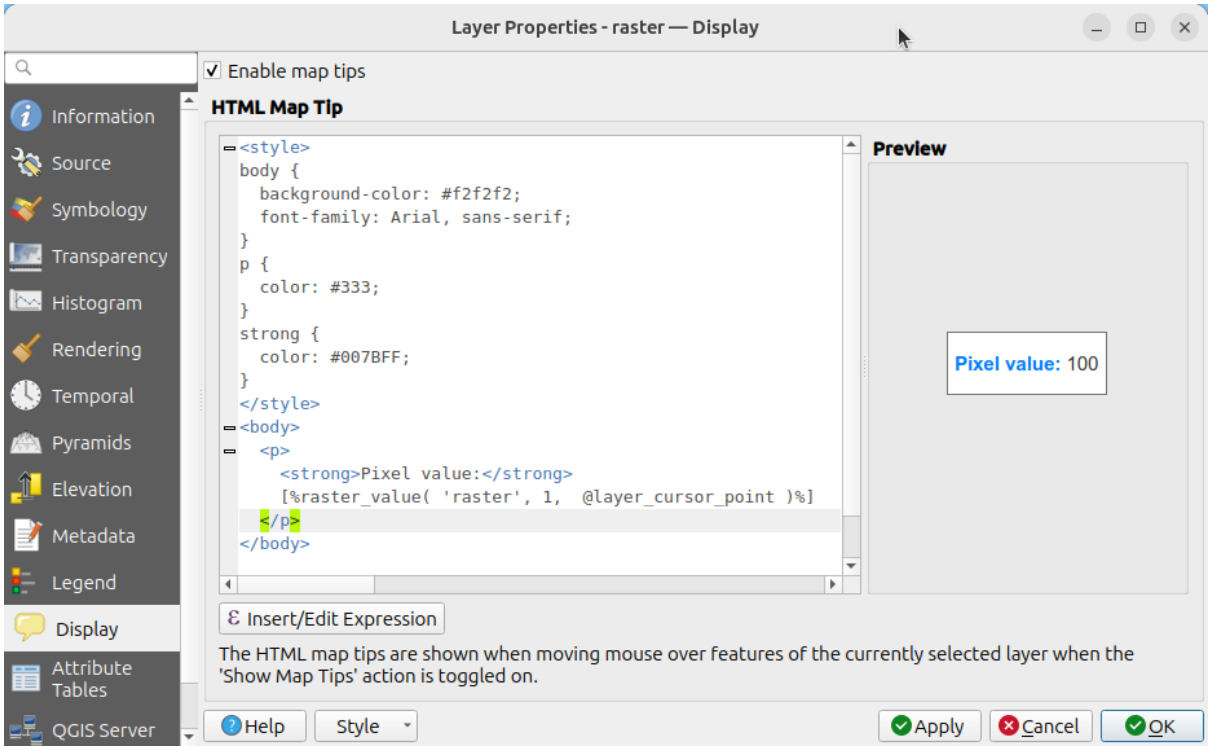
13.1.13 Display Properties

The *Display* tab helps you configure HTML map tips to use for pixels identification:

- *Enable Map Tips* controls whether to display map tips for the layer
- The *HTML Map Tip* provides a complex and full HTML text editor for map tips, mixing QGIS expressions and html styles and tags (multiline, fonts, images, hyperlink, tables, …). You can check the result of your code sample in the *Preview* frame. You can also select and edit existing expressions using the *Insert/Edit Expression* button.

  You might look for expressions located in *Rasters* group or the `@layer_cursor_point` variable in the *Expressions* dialog.

> **Pastaba**
>
> **Understanding the *Insert/Edit Expression* button behavior**
>
> If you select some text within an expression (between „[%“ and „%]“), or if no text is selected but the cursor is inside an expression, the whole expression will be automatically selected for editing. If the cursor or a selected text is outside an expression, the dialog opens with the selection.



13.23 Fig. : Map tips with raster layer

To display map tips:

1. Select the menu option *View* ► *Show Map Tips* or click on the Show Map Tips icon of the *Attributes Toolbar*.
2. Make sure that the layer you target is active and has the *Enable Map Tips* property checked.
3. Move over a pixel, and the corresponding information will be displayed over.

Map tip is a cross-layer feature meaning that once activated, it stays on and applies to any map tip enabled layer in the project until it is toggled off.

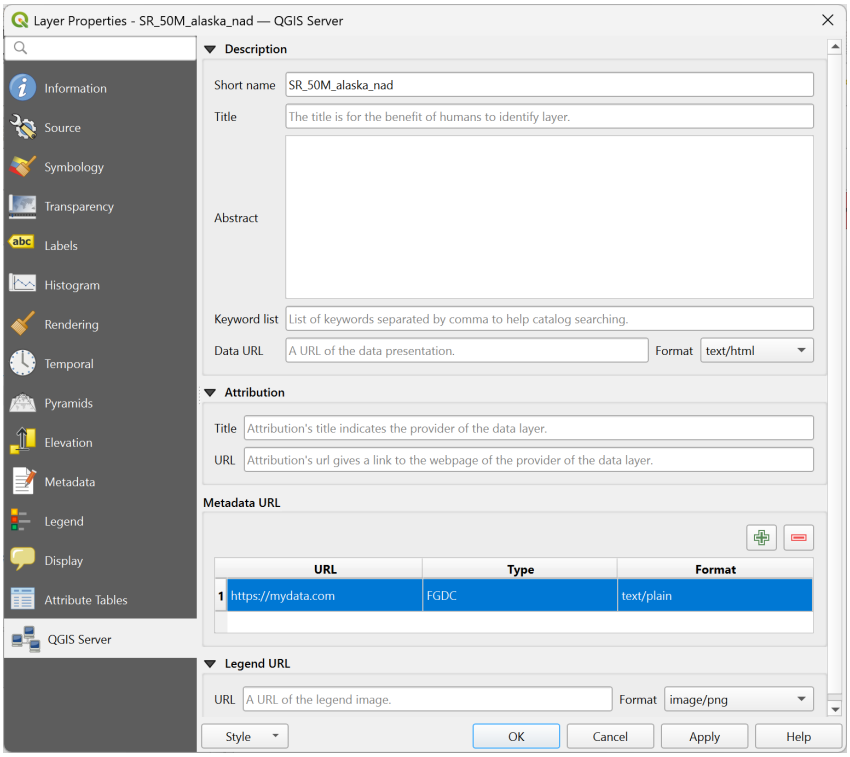
13.1.14 QGIS serverio savybės

The *QGIS Server* tab helps you configure settings of the data when published by QGIS Server. The configuration concerns:

- *Description*: provides information to describe the data, such as *Short name*, *Title*, *Abstract*, a *Keyword list*, and a *Data URL* whose *Format* can be in `text/html`, `text/plain` or `application/pdf`.
- *Attribution*: a *Title* and *URL* to identify who provides the data
- *Metadata URL*: a list of *URL* for the metadata that can be of `FGDC` or `TC211` *Type*, and in `text/plain` or `text/xml` *Format*
- *Legend URL*: a *URL* for the legend, in either `image/png` or `image/jpeg` *Format*

Pastaba

When the raster layer you want to publish is already provided by a web service, further *properties* are available for setting.



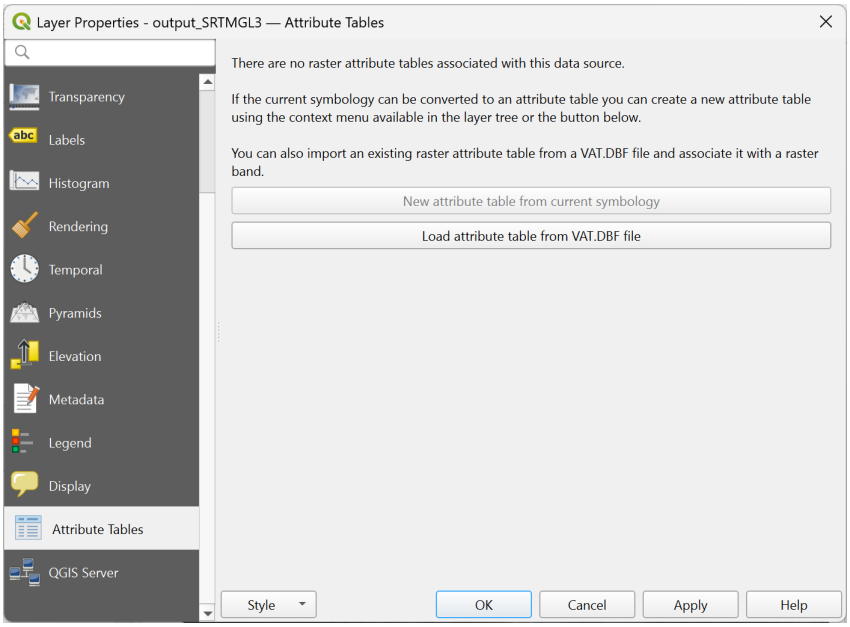
13.24 Fig. : QGIS Server in Raster Properties

13.1.15 Attribute Tables

Similar to vector layers, raster layers in QGIS can have associated Raster Attribute Tables (RATs) that store additional information about raster pixel values.

A Raster Attribute Table links raster values (or value ranges) to descriptive attributes such as class names, pixel counts, and colors. RATs are commonly used for classified rasters, color tables, and histogram information, helping QGIS interpret how raster values should be displayed and classified.

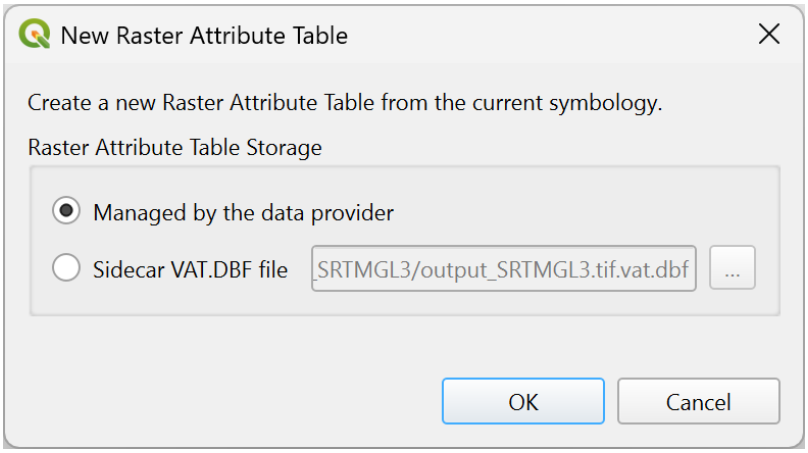
The *Attribute Tables* tab allows you to view and edit raster attribute data. To access the attribute table of a raster layer, right-click the layer in the layers panel, open *Properties…* and select *Attribute Tables*.



13.25 Fig. : Accessing the raster attribute table

When opening the *Attribute Tables* for a raster layer that does not have an associated Raster Attribute Table, QGIS displays a notice informing that no attribute table is currently linked to the data source. In this state, the interface provides two options to define and associate a new attribute table:

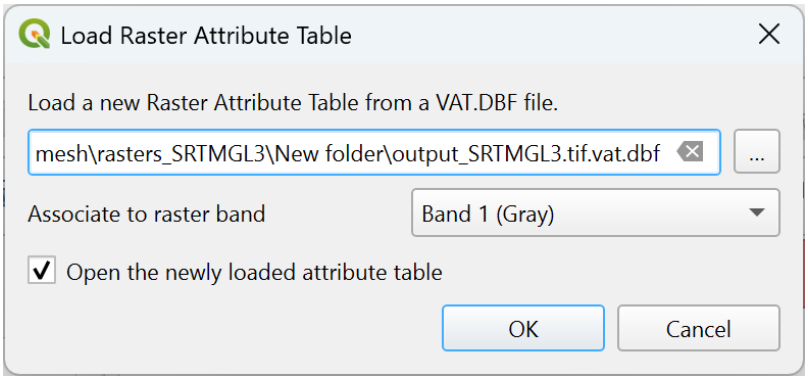
- *New attribute table from current symbology*: will appear if your raster layer has a classified symbology applied. Otherwise, this option is disabled. Applying classified symbology also triggers an option in the layer context menu to *Create Raster Attribute Table*. Click this button and set the following options in the *New Raster Attribute Table* dialog.
  - *Managed by the data provider*: the attribute table will be saved and managed by the data provider (if supported), overwriting any existing table for the raster band used by the current style. Depending on the data provider, the attribute table will be embedded in the main raster file or saved into a sidecar file managed by the data provider.
  - *Sidecar VAT.DBF file*: saves the attribute table into a sidecar VAT.DBF file. The resulting file will not be associated with any particular band.



13.26 Fig. : Create new attribute table from current symbology

- *Load attribute table from VAT.DBF file*: retrieves the attribute table from an external DBF file. *Load Raster Attribute Table* dialog will appear to let you browse and select the DBF file and associate it with the raster

band. The action also can be done by right-clicking the raster layer in the layers panel and selecting *Load Raster Attribute Table from VAT.DBF*. *Open the newly loaded attribute table* allows you to directly open the attribute table editor after loading the table.



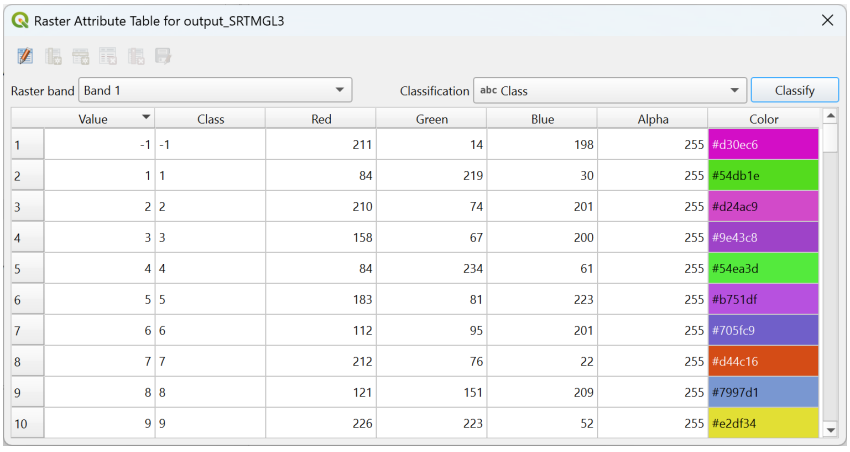
13.27 Fig. : Load attribute table from VAT.DBF file

Editing raster attribute tables

When set, a raster attribute table can be accessed from the *Attribute tables* tab of the layer properties dialog or by selecting *Open raster attribute table* from the layer contextual menu in *Layers* panel. This dialog shows a table whose rows correspond to the different pixel values or value ranges classified in a given band of the raster layer. Depending on the type of classification, a row is described by:

- for an exact values classification (e.g., paletted/unique values): the band name and the applied *Color*. Additional fields of the *Red*, *Green*, *Blue* and *Opacity* of the color are displayed.
- for range values classification (e.g., singleband pseudocolor): the range of values defined by their *Min*, *Max*, the associated label (*Class*) and color ramp interpolated over the values (*Color*). Additional fields of the minimum and maximum of *Red*, *Green*, *Blue* and *Opacity* of the color ramp are displayed.

You can also customize the raster attribute table by adding new fields for classification and group different value ranges accordingly, modifying value ranges and their associated colors or labels, adding new rows, … The new attribute table can be used to update the current layer symbology.



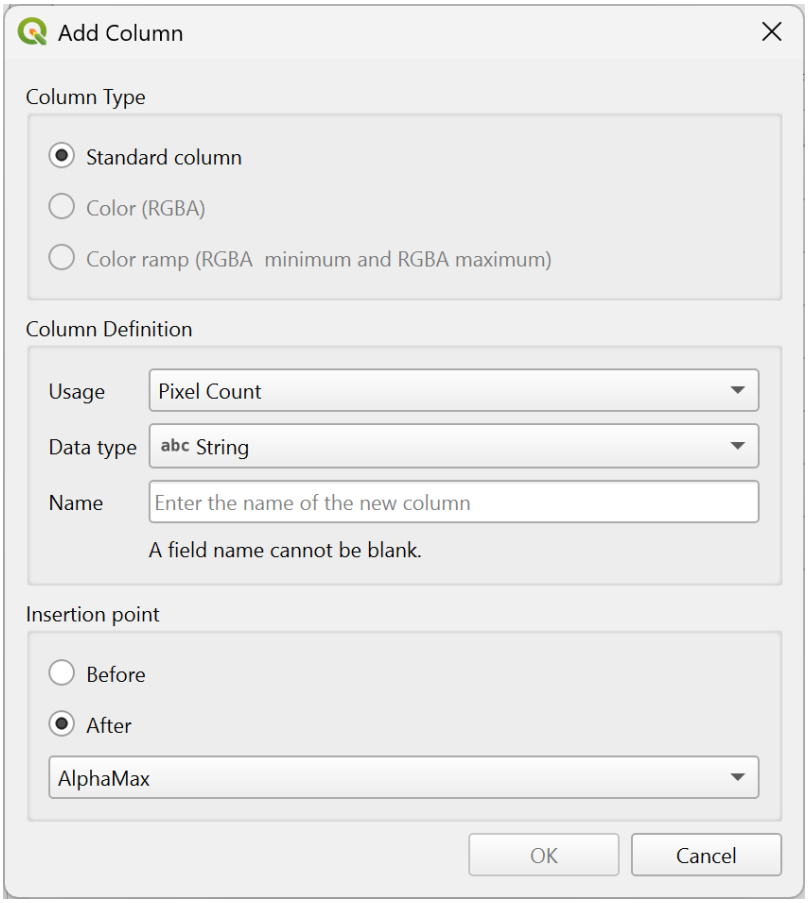
	Value	Class	Red	Green	Blue	Alpha	Color
1	-1	-1	211	14	198	255	#d30ec6
2	1	1	84	219	30	255	#54db1e
3	2	2	210	74	201	255	#d24ac9
4	3	3	158	67	200	255	#9e43c8
5	4	4	84	234	61	255	#54ea3d
6	5	5	183	81	223	255	#b751df
7	6	6	112	95	201	255	#705fc9
8	7	7	212	76	22	255	#d44c16
9	8	8	121	151	209	255	#7997d1
10	9	9	226	223	52	255	#e2df34

13.28 Fig. : Raster Attribute Table

At the top of the dialog, a set of tools allows to edit and save changes to the attribute table:

- Edit Attribute Table turns the table into edit mode. If no specific row or field is selected, only the Add Column… option is available.

- Add Column… opens the *Add Column* dialog to add a new column to the raster attribute table. The dialog lets you define:
  - *Column Type*, which can be:
    - *Standard column*
    - *Color RGBA*
    - *Color ramp (RGBA minimum and RGBA maximum)*
  - *Column Definition*, including:
    - *Name* of the column (mandatory)
    - *Usage*, which can be:
      - *Pixel Count*: stores the number of pixels belonging to a given value or value range.
      - *Generic*: general-purpose column with no predefined meaning.
      - *Name*: stores the class name or label associated with a raster value or range.
    - *Data type*, which can be:
      - *String*
      - *Integer*
      - *Long Integer*
      - *Double*
  - *Insertion point*, defining whether the new column is inserted *Before* or *After* a selected existing column.



13.29 Fig. : Add Column dialog

- *Add Row…* opens the *Add Row* dialog to add a new row to the raster attribute table. Choose the *Insertion point* as *Before current row* or *After current row*.
- Remove Row removes the selected row from the raster attribute table.
- Remove Column removes the selected column from the raster attribute table.
- Save Changes saves any modifications made to the raster attribute table.

The next row provides options for controlling the displayed information in the table:

- *Raster band*: allows selecting which raster band the attribute table applies to. Raster layers may contain multiple bands (e.g., multispectral data). Each band can have its own Raster Attribute Table.
- *Classification*: allows selecting the fields used for classifying the raster layer. Keep in mind that the existing symbology for the raster will be replaced by a new symbology from the attribute table and any unsaved changes to the current symbology will be lost. Make sure to save your symbology style if needed before applying a classification.

13.1.16 Identify raster cells

The *identify features* tool allows you to get information about specific points in a raster layer.

To use the *Identify features* tool:

1. Select the raster layer in the Layers panel.
2. Click on the *Identify features* tool in the toolbar or press `Ctrl+Shift+I`.
3. Click on the point in the raster layer that you want to identify. The pixel will get highlighted.

The Identify Results panel will open in its default `Tree` view and display information about the clicked point. Formatting of the results vary depending on the provider of the layer. For example:

- For a local raster layer: below the name of the layer, you have on the left the band(s) of the clicked pixel, and on the right their respective value.
- For a remote layer such as WMS, a *Format* menu allows you to select whether the information should be displayed as *HTML*, *Feature* or *Text*.

These values can also be rendered (from the *View* menu located at the bottom of the panel) in:

- a `Table` view - organizes the information about the identified features and their values in a table.
- a `Graph` view - organizes the information about the identified features and their values in a graph.

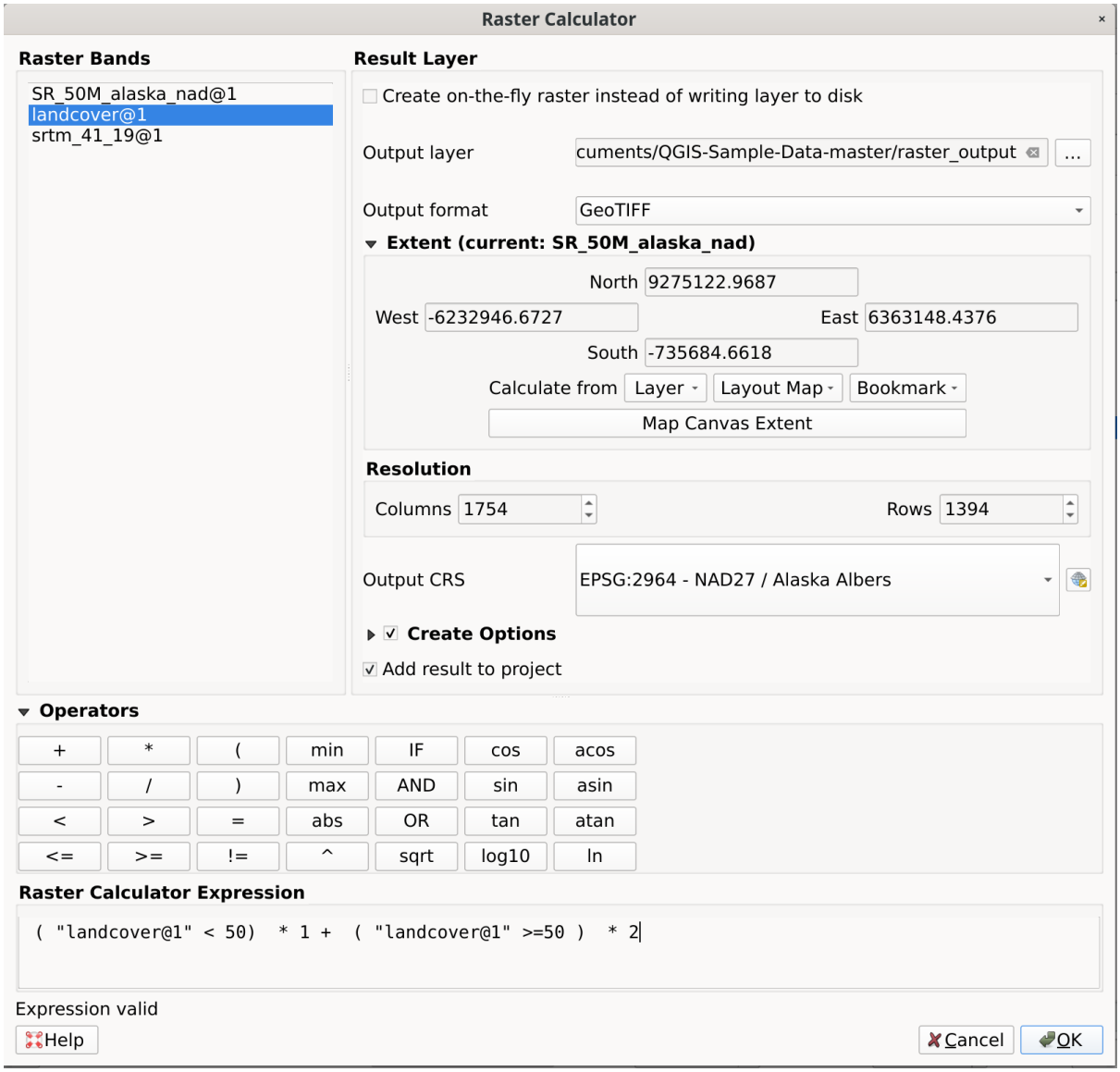
Under the pixel attributes, you will find the *Derived* information, such as:

- `X` ir `Y` koordinačių reikšmės paspaustame taške
- Column and row of the point clicked (pixel) when compatible

13.2 Raster Analysis

13.2.1 Raster Calculator

The *Raster Calculator* in the *Raster* menu allows you to perform calculations on the basis of existing raster pixel values (see Fig. 13.30). The results are written to a new raster layer in a GDAL-supported format.



13.30 Fig. : Raster Calculator

The *Raster bands* list contains all loaded raster layers that can be used. To add a raster to the raster calculator expression field, double click its name in the Fields list. You can then use the operators to construct calculation expressions, or you can just type them into the box.

In the *Result layer* section, you will need to define an output layer. You can:

- *Create on-the-fly raster instead of writing layer to disk*:
  - If unchecked, the output is stored on the disk as a new plain file. An *Output layer* path and an *Output format* are required.
  - If checked, a virtual raster layer, i.e. a raster layer defined by its URI and whose pixels are calculated on-the-fly, is created. It's not a new file on disk; the virtual layer is still connected to the rasters used in the calculation meaning that deleting or moving these rasters would break it. A *Layer name* can be provided, otherwise the calculation expression is used as such. Removing the virtual layer from the project deletes it, and it can be made persistent in file using the layer *Export* ► *Save as…* contextual menu.
- Define the *Spatial extent* of the calculation using the *extent selector* widget.
- Set the *Resolution* of the layer using columns and rows number. If the input layer has a different resolution, the values will be resampled with the nearest neighbor algorithm.

- Enable the *Create Options* to define additional raster creation options. These options let you control how the output file is structured and compressed, including format, compression method, and other driver-specific settings. See more at *Raster driver options*.
- With the *Add result to project* checkbox, the result layer will automatically be added to the legend area and can be visualized. Checked by default for virtual rasters.

The *Operators* section contains all available operators. To add an operator to the raster calculator expression box, click the appropriate button. Mathematical calculations (+, −, *, …) and trigonometric functions (`sin`, `cos`, `tan`, …) are available. Conditional expressions (=, !=, <, >=, …) return either 0 for false or 1 for true, and therefore can be used with other operators and functions.

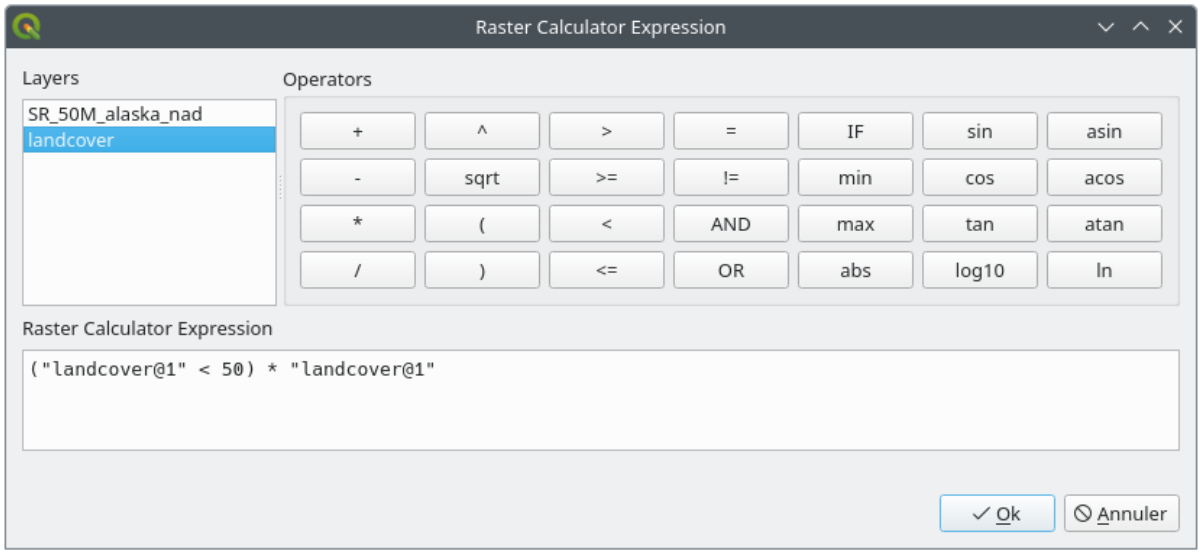
Taip pat žiūrėkite

Raster calculator and *Raster calculator (virtual)* algorithms

Raster calculator expression

The dialog

The *Raster calculator expression* dialog provides means to write expressions for pixels calculations between a set of raster layers.



13.31 Fig. : Raster Expression Calculator

- *Layers*: Shows the list of all raster layers loaded in the legend. These can be used to fill the expression box (double click to add). Raster layers are referred by their name and the number of the band: `layer_name@band_number`. For instance, the first band from a layer named `DEM` will be referred as `DEM@1`.
- *Operators*: contains a number of calculation operators for pixels manipulation:
  - Arithmetical: +, −, *, `sqrt`, `abs`, `ln`, …
  - Trigonometric: `sin`, `cos`, `tan`, …
  - Comparison: =, !=, <, >=, …
  - Logical: `IF`, `AND`, `OR`, `(`, `)`
  - Statistical: `min`, `max`

To add an operator to the raster calculator expression box, click the appropriate button.

- *Raster calculator expression* is the area in which the expression is composed

Pavyzdžiai

Convert elevation values from meters to feet

Creating an elevation raster in feet from a raster in meters, you need to use the conversion factor for meters to feet: 3.28. The expression is:

```
"elevation@1" * 3.28
```

Using a mask

If you want to mask out parts of a raster – say, for instance, because you are only interested in elevations above 0 meters – you can use the following expression to create a mask and apply the result to a raster in one step.

```
("elevation@1" >= 0) * "elevation@1"
```

In other words, for every cell greater than or equal to 0 the conditional expression evaluates to 1, which keeps the original value by multiplying it by 1. Otherwise the conditional expression evaluates to 0, which sets the raster value to 0. This creates the mask on the fly.

Classify a Raster

If you want to classify a raster – say, for instance into two elevation classes, you can use the following expression to create a raster with two values 1 and 2 in one step.

```
("elevation@1" < 50) * 1 + ("elevation@1" >= 50) * 2
```

In other words, for every cell less than 50 set its value to 1. For every cell greater than or equal 50 set its value to 2.

Or you can use the `IF` operator.

```
if ( elevation@1 < 50 , 1 , 2 )
```

SKYRIUS 14

Darbas su tinklelio duomenimis

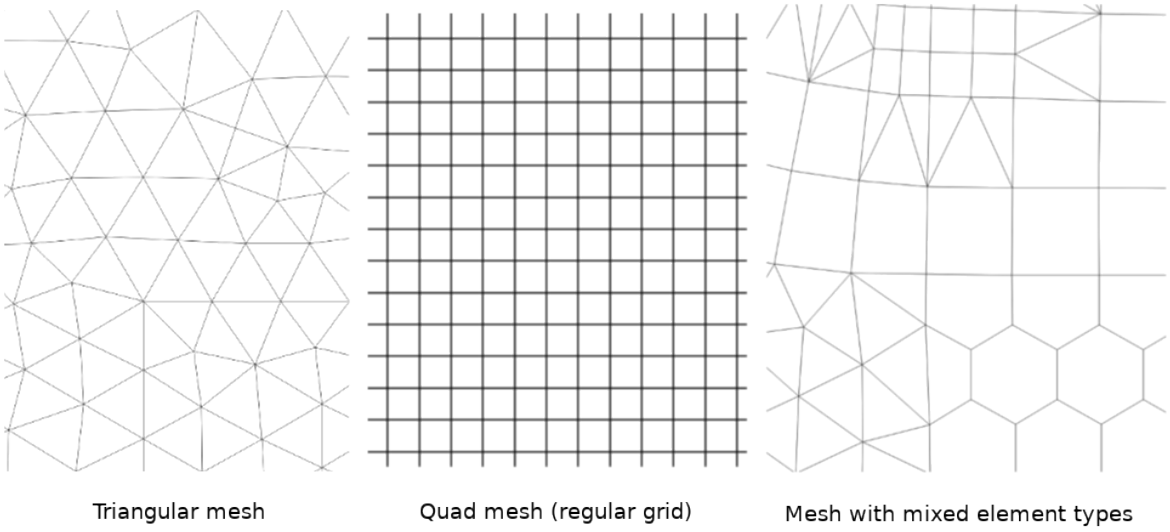
14.1 Kas yra tinklelis?

Tinklelis yra nestruktūrinis tinklas, paprastai su laiko ar kitais komponentais. Erdviniame komponente yra 2D ar 3D erdvės viršūnių, kraštinių ir/ar plokštumų rinkinys:

- **viršūnės** - XY(Z) taškai (sluoksnio koordinačių atskaitos sistemoje)
- **kraštinės** - jungia viršūnių poras
- **plokštumos** - plokštuma yra kraštinių rinkinys, sukuriantis uždarą formą - paprastai trikampį ar keturkampį, rečiau poligonus su daugiau viršūnių

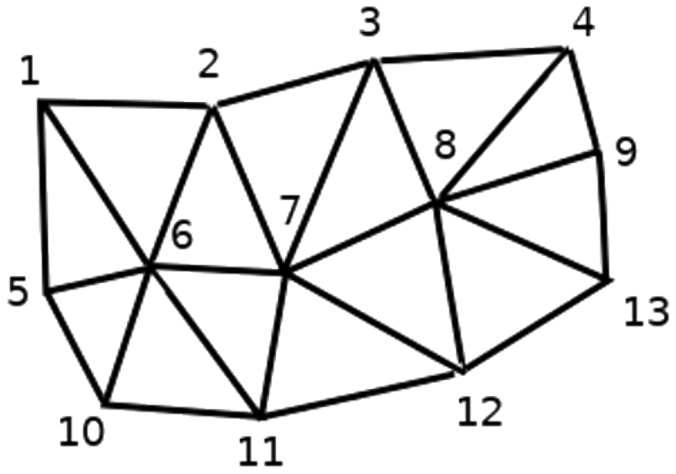
Relying on the above, mesh layers can thus have different types of structure:

- 1D Meshes: consist of vertices and edges. An edge connects two vertices and can have assigned data (scalars or vectors) on it. The 1D mesh network can be for example used for modelling of an urban drainage system.
- 2D meshes: consist of faces with triangles, regular or unstructured quads.
- 3D layered meshes: consist of multiple stacked 2D unstructured meshes each extruded in the vertical direction (levels) by means of a vertical coordinate. The vertices and faces have the same topology in each vertical level. The mesh definition (vertical level extrusion) could in general change in time. The data is usually defined in volume centres or by some parametric function.



14.1 Fig. : Different mesh types

Mesh provides information about the spatial structure. In addition, the mesh can have datasets (groups) that assign a value to every vertex. For example, having a triangular mesh with numbered vertices as shown in the image below:



14.2 Fig. : Triangular grid with numbered vertices

Each vertex can store different datasets (typically multiple quantities), and those datasets can also have a temporal dimension. Thus, a single file may contain multiple datasets.

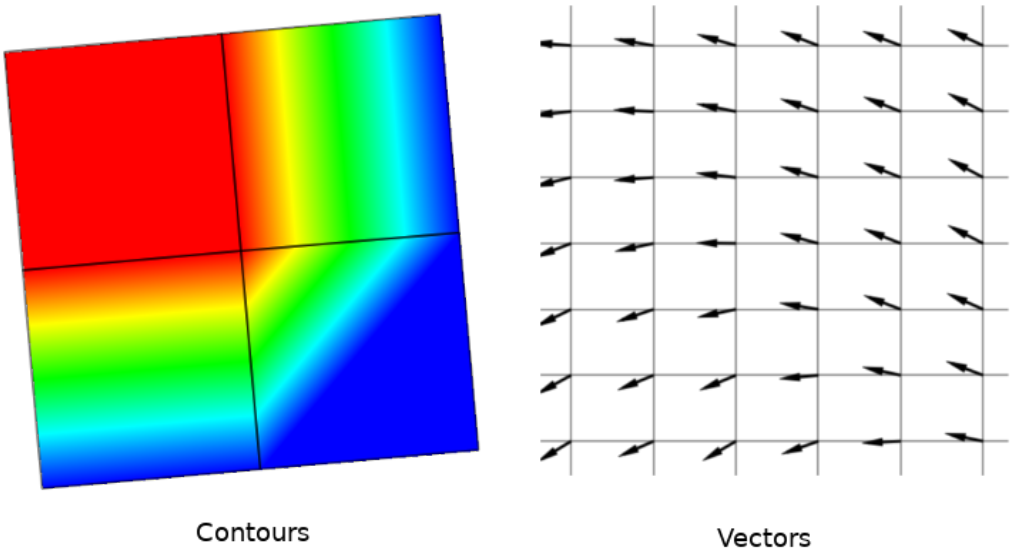
The following table gives an idea about the information that can be stored in mesh datasets. Table columns represent indices of mesh vertices, each row represents one dataset. Datasets can have different datatypes. In this case, it stores wind velocity at 10m at a particular moments in time (t1, t2, t3).

In a similar way, the mesh dataset can also store vector values for each vertex. For example, wind direction vector at the given time stamps:

14.1 Table : Example of mesh dataset

10 metre wind	1	2	3	...
10 metre speed at time=t1	17251	24918	32858	…
10 metre speed at time=t2	19168	23001	36418	…
10 metre speed at time=t3	21085	30668	17251	…
…	…	…	…	…
10m wind direction time=t1	[20,2]	[20,3]	[20,4.5]	…
10m wind direction time=t2	[21,3]	[21,4]	[21,5.5]	…
10m wind direction time=t3	[22,4]	[22,5]	[22,6.5]	…
…	…	…	…	…

We can visualize the data by assigning colors to values (similarly to how it is done with *Singleband pseudocolor* raster rendering) and interpolating data between vertices according to the mesh topology. It is common that some quantities are 2D vectors rather than being simple scalar values (e.g. wind direction). For such quantities it is desirable to display arrows indicating the directions.



14.3 Fig. : Possible visualisation of mesh data

14.2 Supported formats

QGIS accesses mesh data using the MDAL drivers, and natively supports a variety of formats. Whether QGIS can edit a mesh layer depends on the format and the mesh structure type.

To load a mesh dataset into QGIS, use the *Mesh* tab in the *Data Source Manager* dialog. Read *Tinklelio sluoksnio įkėlimas* for more details.

14.3 Mesh Dataset Properties

The *Layer Properties* dialog for a mesh layer provides general settings to manage dataset groups of the layer and their rendering (active dataset groups, symbology, 2D and 3D rendering). It also provides information about the layer.

Norėdami pasiekti *Sluoksnio savybių* dialogą:

- Skydelyje *Sluoksniai* du kartus spauskite sluoksnį arba spauskite dešiniu pelės mygtuku ir tada iššokančiame meniu parinkite *Savybės...*;
- Parinkus sluoksnį eikite į meniu *Sluoksniai* ► *Sluoksnio savybės...*.

The mesh *Layer Properties* dialog provides the following sections:

14.2 Table : Tabs of the Mesh Layer Properties

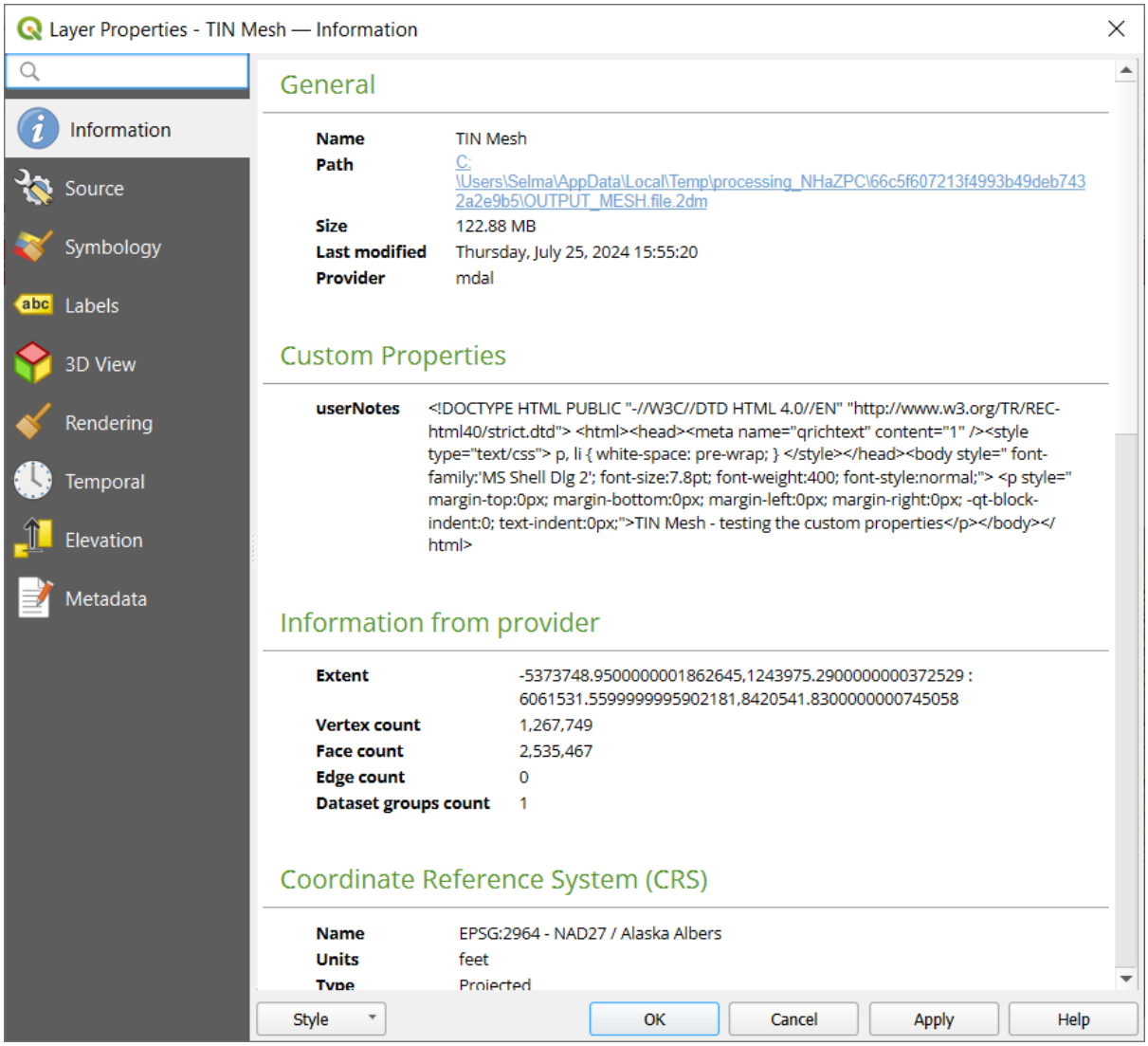
Information	*Source*	*Symbology* ([1])
Labels ([1])	*3D View* ([1])	*Rendering*
Temporal	*Elevation*	*Metadata*

Pastaba

Most of the properties of a mesh layer can be saved to or loaded from a `.qml` using the *Style* menu at the bottom of the dialog. More details at *Savo stilių tvarkymas*.

[1] Also available in the *Layer styling panel*

14.3.1 Informacinės savybės



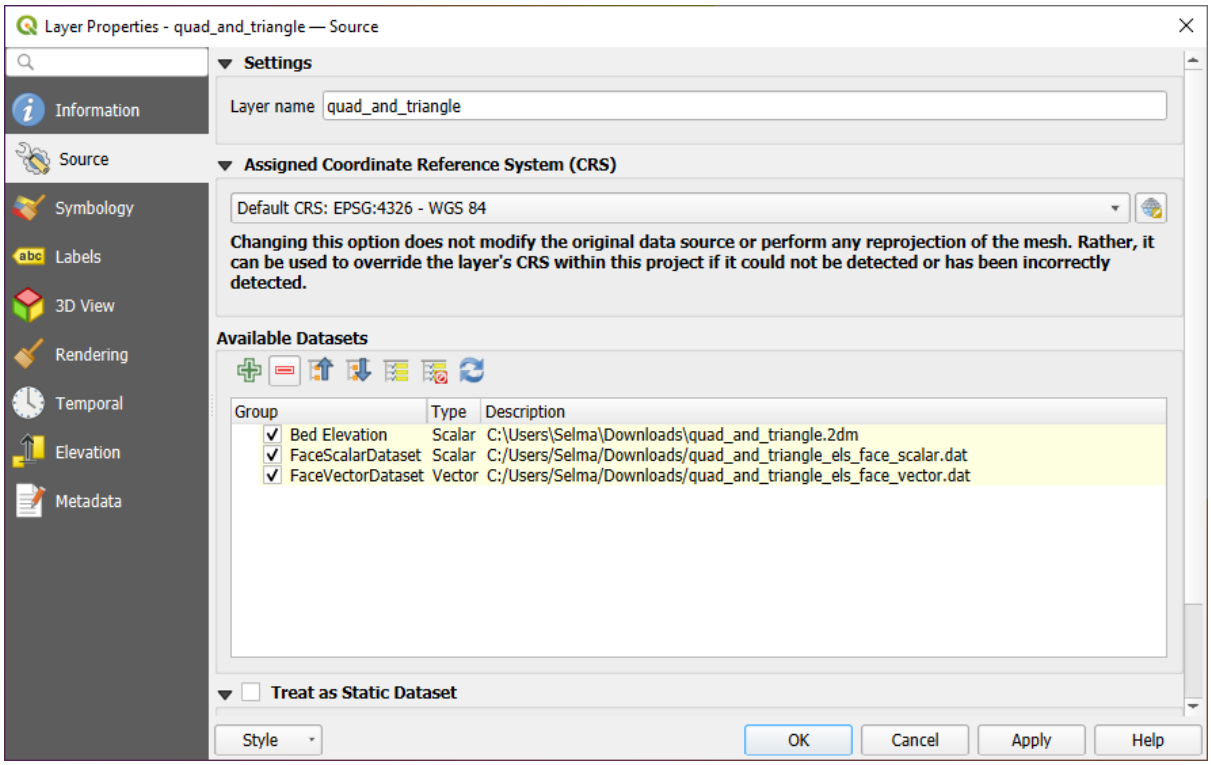
14.4 Fig. : Mesh Layer Information Properties

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata on the current layer. Provided information are:

- general such as name in the project, source path, list of auxiliary files, last save time and size, the used provider
- custom properties, used to store in the active project additional information about the layer. Default custom properties may include *layer notes*. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- based on the provider of the layer: extent, vertex, face, edges and/or dataset groups count
- the Coordinate Reference System: name, units, method, accuracy, reference (i.e. whether it's static or dynamic)
- extracted from filled *metadata*: access, extents, links, contacts, history…

14.3.2 Source Properties

The *Source* tab displays basic information about the selected mesh, including:



14.5 Fig. : Mesh Layer Source Properties

- the layer name to display in the *Layers* panel
- setting the Coordinate Reference System: Displays the layer's *Assigned Coordinate Reference System (CRS)*.

  You can change the layer's CRS by selecting a recently used one in the drop-down list or clicking on *Select CRS* button (see *Koordinačių atskaitos sistemos parinkiklis*). Use this process only if the CRS applied to the layer is wrong or if none was applied.
- The *Available datasets* frame lists all the dataset groups (and subgroups) in the mesh layer, with their type and description in a tree view. Both regular datasets (i.e. their data is stored in the file) and virtual datasets (which are *calculated on the fly*) are listed.
  - Use the Assign Extra Dataset to Mesh button to add more groups to the current mesh layer. You can add dataset group to a mesh layer with the same name, but not from the same URI. Dataset group names are automatically renamed to „Original Name_Number".
  - Use Remove Extra Dataset from Mesh to remove additional datasets groups from the mesh layer. Note that only dataset groups not associated with the mesh source file can be removed.
  - Collapse all and Expand all the dataset tree, in case of embedded groups
  - If you are interested in few datasets, you can uncheck the others and make them unavailable in the project
  - Double-click over a name and you can rename the dataset.
  - Reset to defaults: checks all the groups and renames them back to their original name in the provider.
  - Right-click over a virtual dataset group and you can:
    - *Remove dataset group* from the project

    * *Save dataset group as…* a file on disk, to any supported format. The new file is kept assigned to the current mesh layer in the project.

- Checking the *Treat as static dataset* group allows to ignore the *map temporal navigation* properties while rendering the mesh layer. For each active dataset group (as selected in *Symbology* ► *Datasets* tab), you can:
  - set to *None*: the dataset group is not displayed at all
  - *Display dataset*: e.g., for the „bed elevation“ dataset which is not time aware
  - extract a particular date time: the dataset matching the provided time is rendered and stays fixed during map navigation.

14.3.3 Symbology Properties

Click the *Symbology* button to activate the dialog. Symbology properties are divided into several tabs:

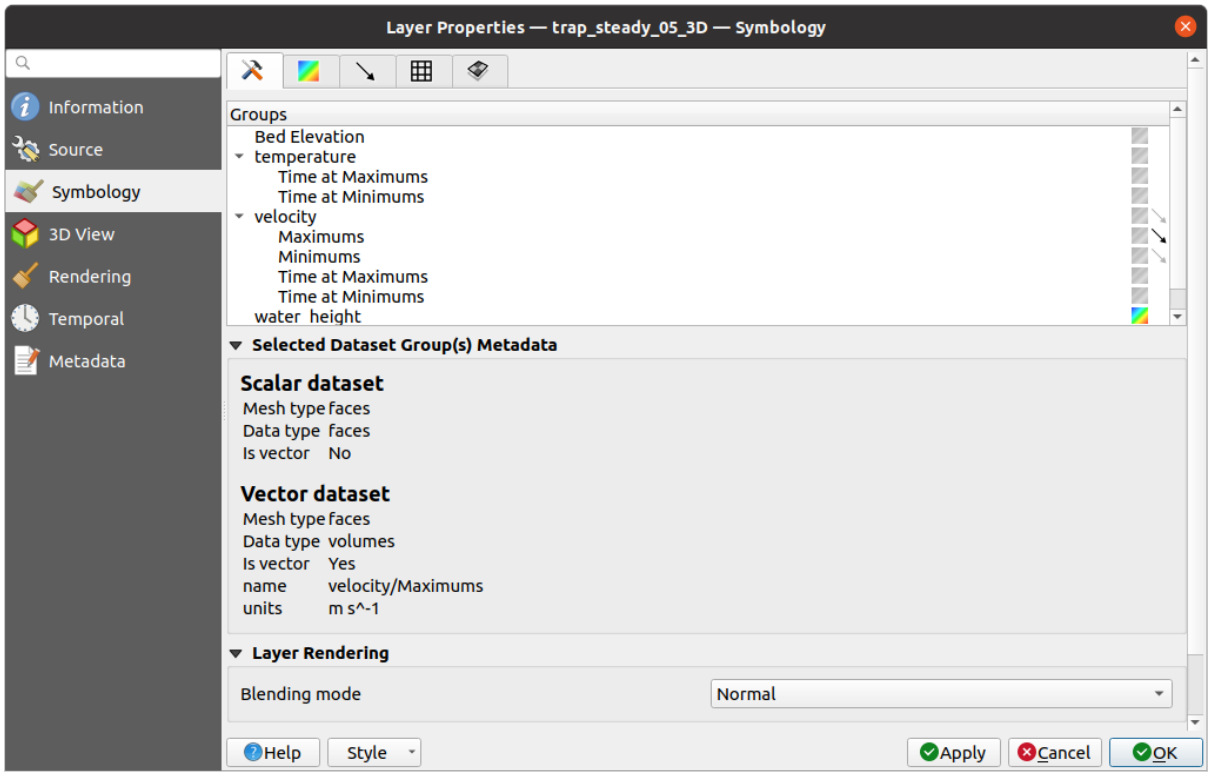
- *Datasets*
- *Contours*
- *Vectors*
- *Rendering*
- *Stacked mesh averaging method*

Datasets

The tab Datasets is the main place to control and set which datasets will be used for the layer. It presents the following items:

- *Groups* available in the mesh dataset, with whether they provide:
  - scalar dataset
  - or vector dataset: by default, each vector dataset has a scalar dataset representing its magnitude automatically generated.

  Click on the icon next to the dataset name to select the group and type of data to represent.
- *Selected dataset group(s) metadata*, with details on:
  - the mesh type: edges or faces
  - the data type: vertices, edges, faces or volume
  - whether it's of vector type or not
  - the original name in the mesh layer
  - the unit, if applicable
- *blending mode* available for the selected datasets.



14.6 Fig. : Mesh Layer Datasets

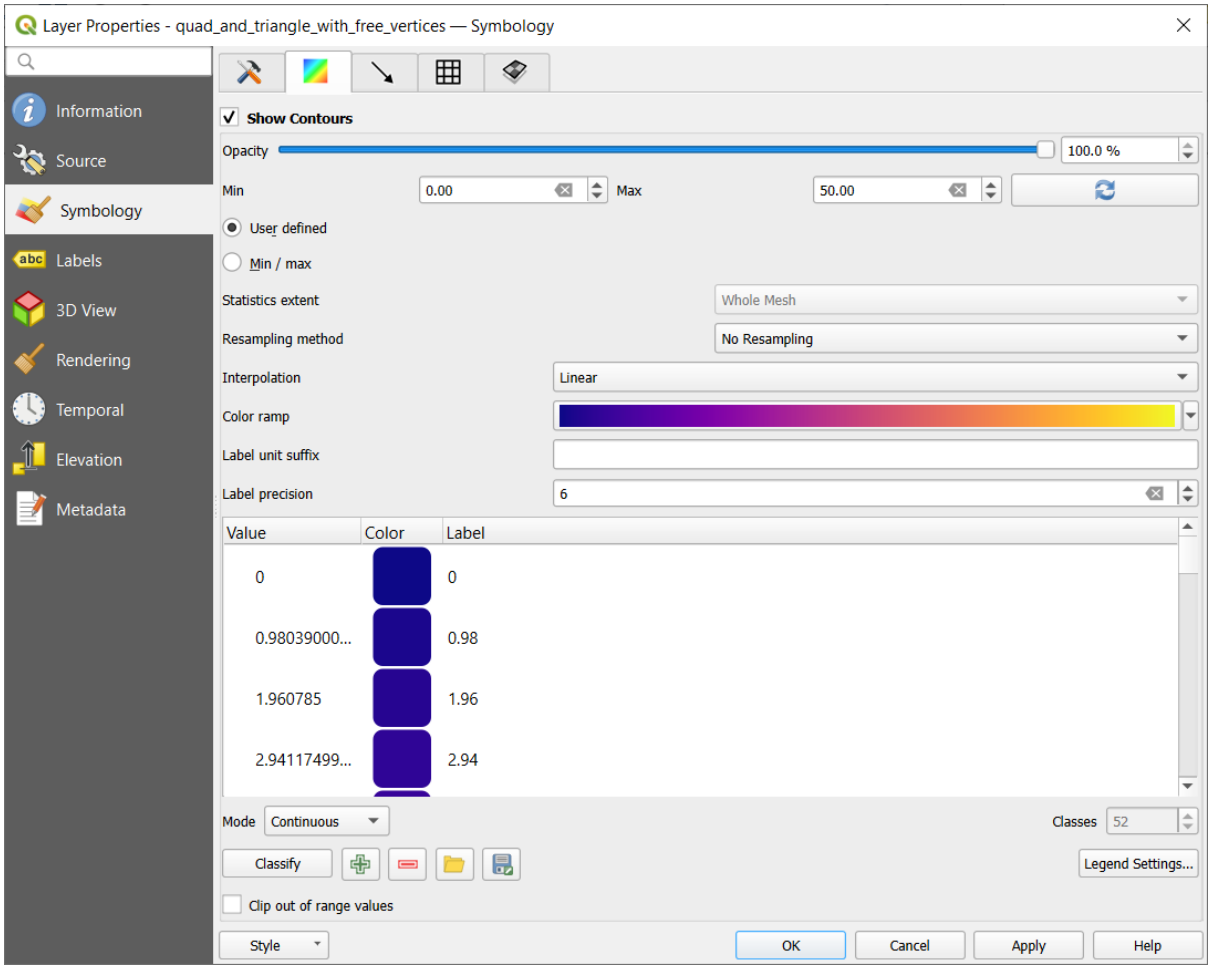
You can apply symbology to the selected vector and/or scalar group using the next tabs.

Contours Symbology

Pastaba

The Contours tab can be activated only if a scalar dataset has been selected in the *Datasets* tab.

In the Contours tab you can see and change the current visualization options of contours for the selected group, as shown in Fig. 14.7 below:



14.7 Fig. : Styling Contours in a Mesh Layer

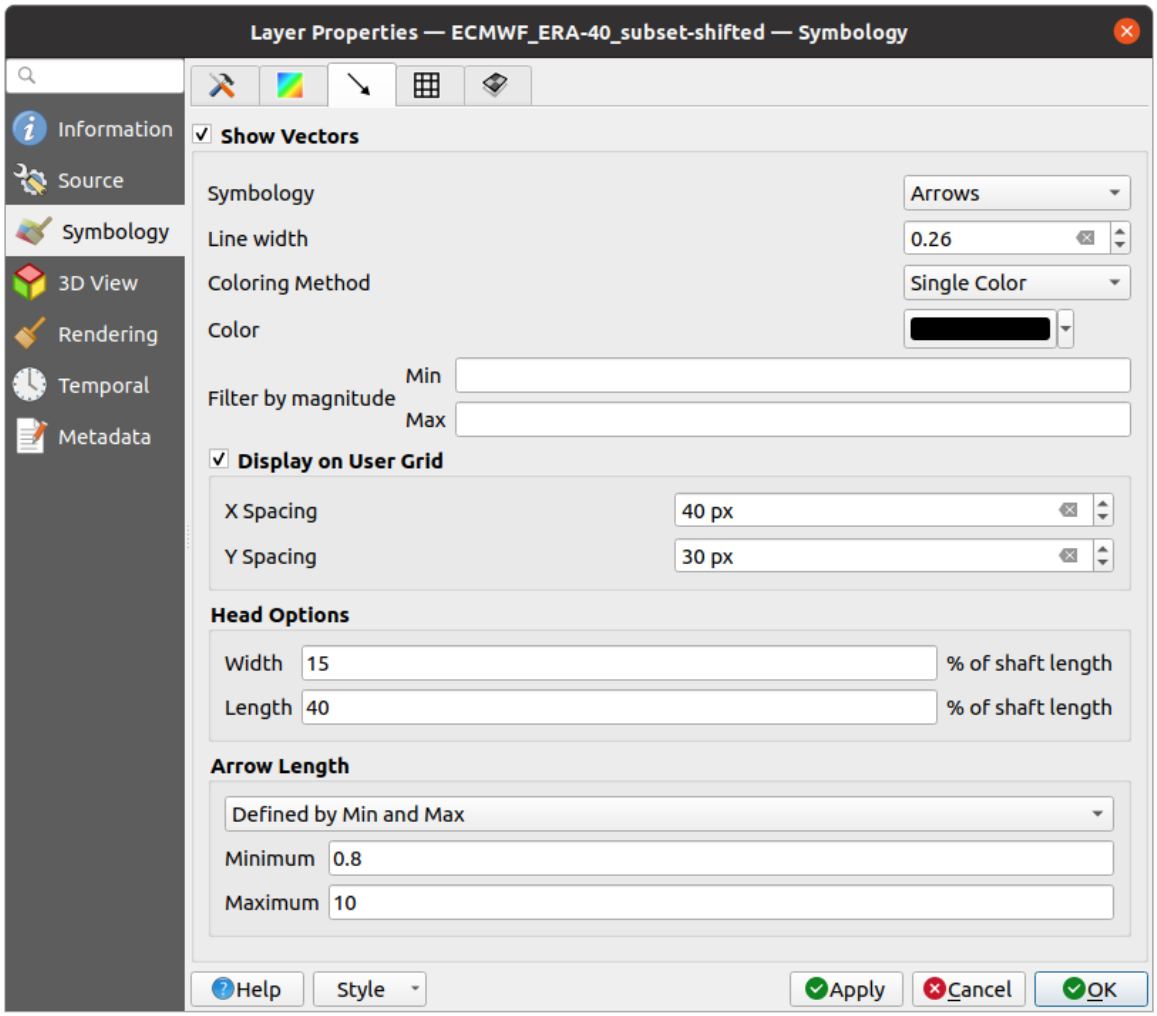
- For 1D mesh, set the *Stroke width* of the edges. This can be a fixed size for the whole dataset, or vary along the geometry (more details with the *interpolated line renderer*)
- Use the slider or the spinbox to set the *Opacity* of the current group, if of a 2D mesh type.
- *User Defined* allows you to enter the range of values you want to represent on the current group: use Load to fetch the min and max values of the current group or enter custom values if you want to exclude some.
- Select *Min/Max* to set the renderer's minimum and maximum values based on the chosen extent.
- *Statictics extent* can be *Whole mesh*, *Current canvas* or *Updated canvas*. *Updated canvas* means that min/max values used for the rendering will change with the canvas extent (dynamic stretching).
- For 2D/3D meshes, select the *Resampling method* to interpolate the values on the surrounding vertices to the faces (or from the surrounding faces to the vertices) using the *Neighbour average* method. Depending on whether the dataset is defined on the vertices (respectively on the faces), QGIS defaults this setting to *None* (respectively *Neighbour average*) method in order to use values on vertices and keep the default rendering smooth.
- Classify the dataset using the *color ramp shader* classification.

Vectors Symbology

Pastaba

The Vectors tab can be activated only if a vector dataset has been selected in the *Datasets* tab. Click on the Vectors icon on the right side of the *Datasets* tab.

In the Vectors tab you can see and change the current visualization options of vectors for the selected group, as shown in Fig. 14.8:



14.8 Fig. : Styling Vectors in a Mesh Layer with arrows

Mesh vector dataset can be styled using various types of *Symbology*:

- **Arrows**: vectors are represented with arrows at the same place as they are defined in the raw dataset (i.e. on the nodes or center of elements) or on a user-defined grid (hence, they are evenly distributed). The arrow length is proportional to the magnitude of the arrow as defined in the raw data but can be scaled by various methods.
- **Streamlines**: vectors are represented with streamlines seeded from start points. The seeding points can start from the vertices of the mesh, from a user grid or randomly.
- **Traces**: a nicer animation of the streamlines, the kind of effect you get when you randomly throws sand in the water and see where the sand items flows.

- **Wind barbs**: vectors are represented with wind barbs, a common way to represent wind speed and direction.

Available properties depend on the selected symbology as shown in the following table.

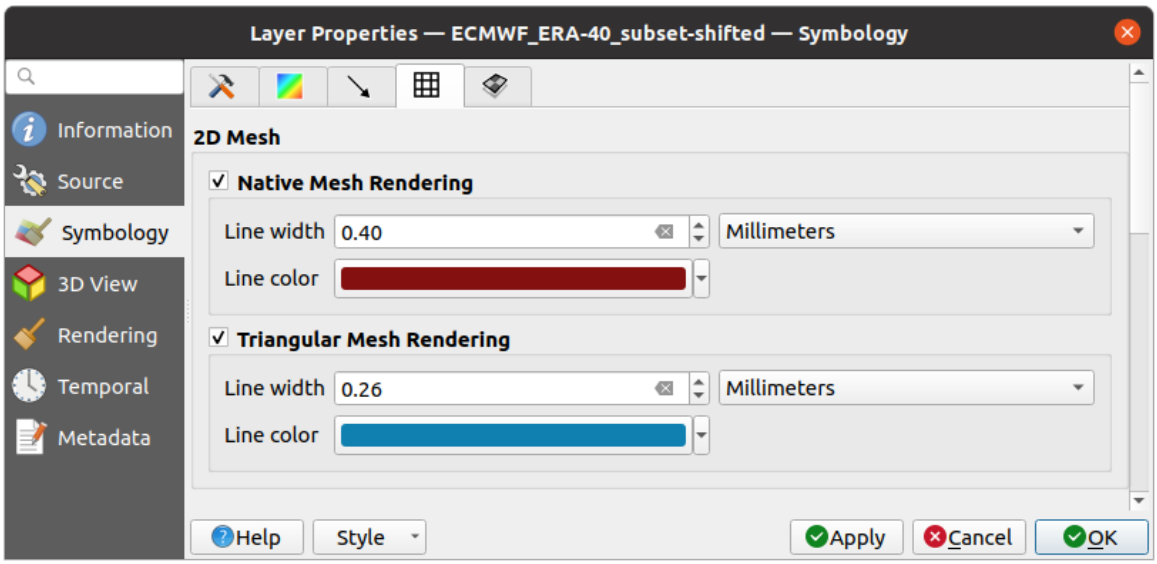
14.3 Table : Availability and meaning of the vectors symbology properties

Label, Description and Properties	Arrow	Streamlines	Traces	Wind Barbs
Line width: Width of the vector representation	☑	☑	☑	☑
Coloring method: • a *Single color* assigned to all vectors • or a variable color based on vectors magnitude, using a *Color ramp shader*	☑	☑	☑	☑
Filter by magnitude: Only vectors whose length for the selected dataset falls between a *Min* and *Max* range are displayed	☑	☑		☑
Display on user grid: Places the vector on a grid with custom *X spacing* and *Y spacing* and interpolates their length based on neighbours	☑	☑		☑
Head options: *Length* and *Width* of the arrow head, as a percentage of its shaft length	☑			
Arrow length: • **Defined by Min and Max**: You specify the minimum and maximum length for the arrows, QGIS will interpolate their size based on the underlying vector's magnitude • **Scale to magnitude**: arrow length is proportional to their vector's magnitude • **Fixed**: all the vectors are shown with the same length	☑			
Streamlines seeding method: • **On mesh/grid**: relies on the user grid to display the vectors • **Randomly**: vector placement is randomly done with respect to a certain density		☑		
Particles count: The amount of „sand" you want to throw into visualisation			☑	
Max tail length: The time until the particle fades out			☑	
Wind Barbs: • **Length**: Specifies the length of the wind barbs. Choose the units from a drop-down menu. • **Data Units**: Specifies the units of wind speed data. Wind barbs encode the wind in knots (nautical miles per hour). Therefore, you should select the units that your data are in, or use a custom *Multiplier* to convert your data units to knots. This ensures that the wind speed is accurately represented.				☑

Rendering

In the tab Rendering tab, QGIS offers possibilities to display and customize the mesh structure. *Line width* and *Line color* can be set to represent:

- the edges for 1D meshes
- For 2D meshes:
  - *Native mesh rendering*: shows original faces and edges from the layer
  - *Triangular mesh rendering*: adds more edges and displays the faces as triangles



14.9 Fig. : 2D Mesh Rendering

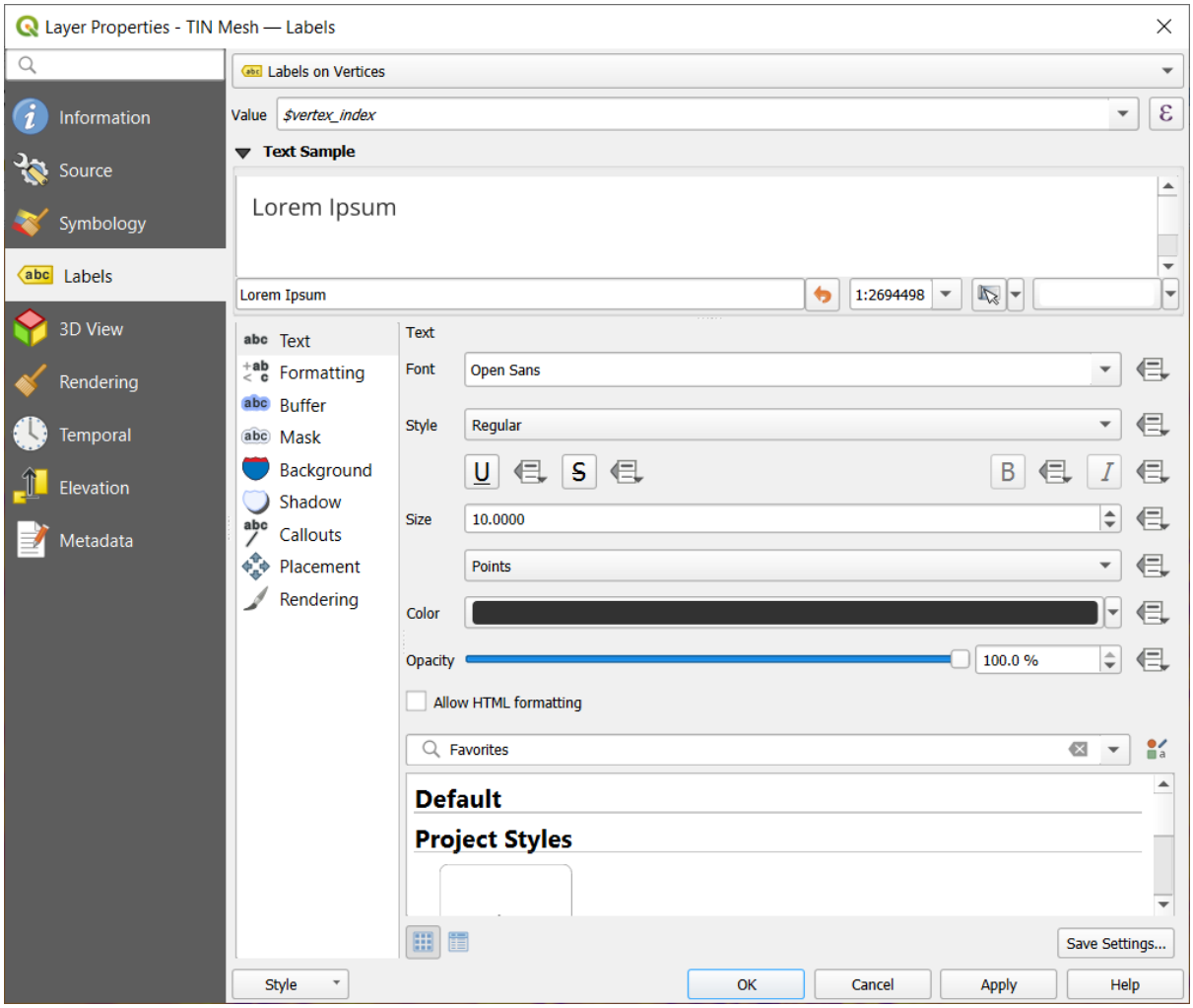
Stacked mesh averaging method

3D layered meshes consist of multiple stacked 2D unstructured meshes each extruded in the vertical direction (`levels`) by means of a vertical coordinate. The vertices and faces have the same topology in each vertical level. Values are usually stored on the volumes that are regularly stacked over base 2d mesh. In order to visualise them on 2D canvas, you need to convert values on volumes (3d) to values on faces (2d) that can be shown in mesh layer. The Stacked mesh averaging method provides different averaging/interpolation methods to handle this.

You can select the method to derive the 2D datasets and corresponding parameters (level index, depth or height values). For each method, an example of application is shown in the dialog but you can read more on the methods at https://fvwiki.tuflow.com/Depth_Averaging_Results.

14.3.4 Labels Properties

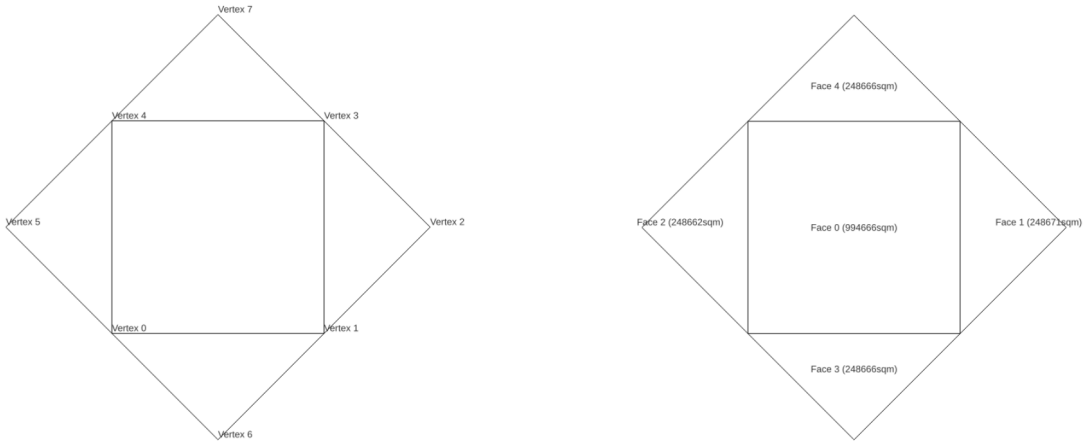
The *Labels* tab offers you dynamic labeling of mesh vertices and native mesh faces based on geometric properties and custom expressions. This dialog can also be accessed from the *Layer Styling* panel.



14.10 Fig. : Mesh Labels Properties

Choose the labeling method from the drop-down list. Available methods are:

- *No labels*: the default value, showing no labels from the layer.
- *Labels on vertices*: labels the mesh vertices as a point feature.
- *Labels on faces*: labels the mesh faces as a polygon feature.



14.11 Fig. : Mesh Labeling methods

Once the labeling method has been selected, open Expression dialog and choose a *mesh function*. You can combine meshes functions with other functions from the *list* to modify the label.

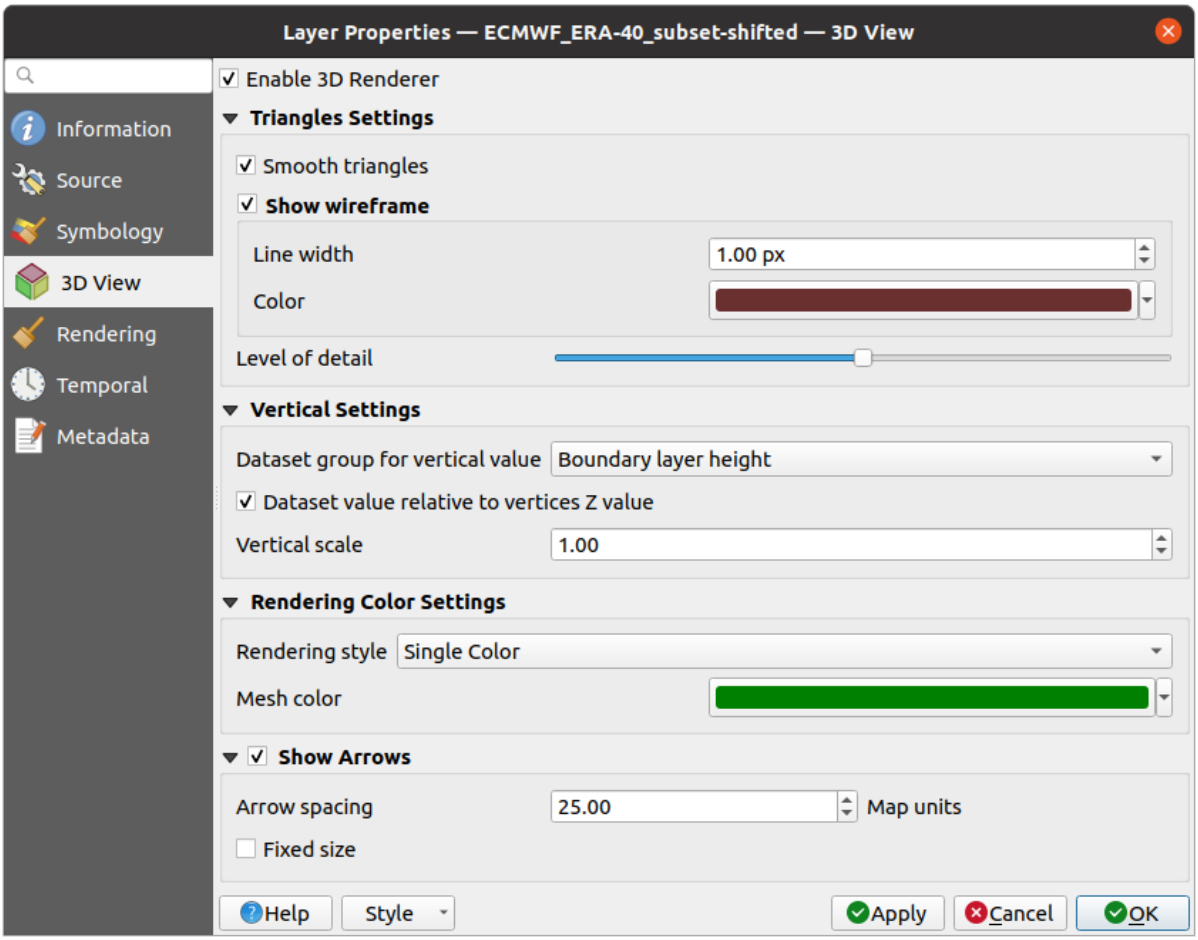
Below are displayed options to customize the labels, under various tabs:

- *Text*
- *Formatting*
- *Buffer*
- *Mask*
- *Background*
- *Shadow*
- *Callouts*
- *Placement*
- *Rendering*

Description of how to set each property is exposed at *Užrašo nustatymas*.

14.3.5 3D View Properties

Mesh layers can be used as *terrain in a 3D map view* based on their vertices Z values. From the *3D View* properties tab, it's also possible to render the mesh layer's dataset in the same 3D view. Therefore, the vertical component of the vertices can be set equal to dataset values (for example, level of water surface) and the texture of the mesh can be set to render other dataset values with color shading (for example velocity).

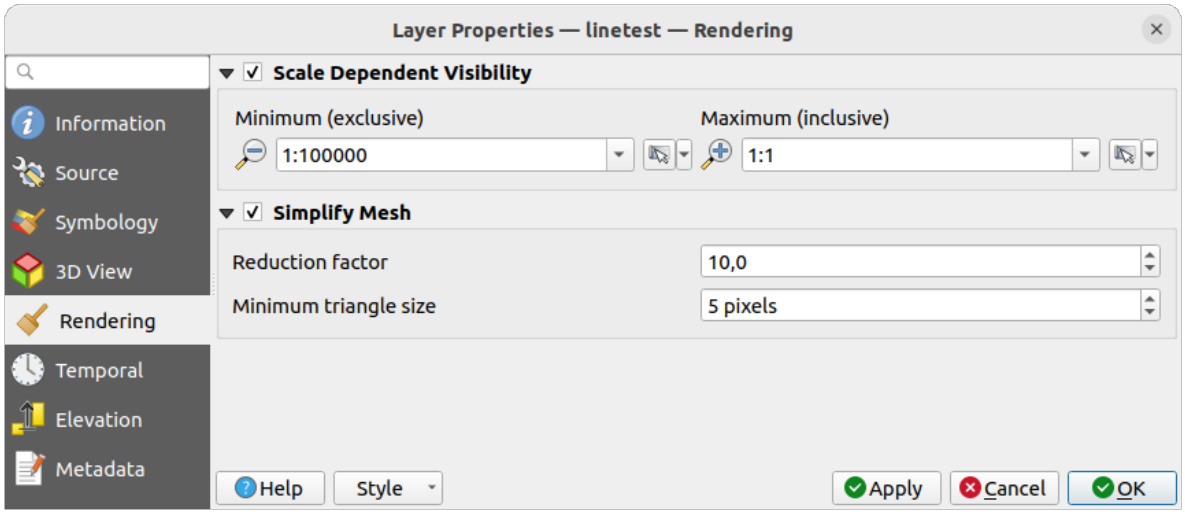


14.12 Fig. : Mesh dataset 3D properties

Check *Enable 3D Renderer* and you can edit following properties:

- Under *Triangle settings*
  - *Smooth triangles*: Angles between consecutive triangles are smoothed for a better 3D rendering
  - *Show wireframe* whose you can set the *Line width* and *Color*
  - *Level of detail*: Controls how *simplified* the mesh layer to render should be. On the far right, it is the base mesh, and the more you go left, the more the layer is simplified and is rendered with less details. This option is only available if the *Simplify mesh* option under the *Rendering* tab is activated.
- *Vertical settings* to control behavior of the vertical component of vertices of rendered triangles:
  - *Dataset group for vertical value*: the dataset group that will be used for the vertical component of the mesh
  - *Dataset value relative to vertices Z value*: whether to consider the dataset values as absolute Z coordinate or relative to the vertices native Z value
  - *Vertical scale*: the scale factor to apply to the dataset Z values
- *Rendering color settings* with a *Rendering style* that can be based on the color ramp shader set in *Contours Symbology* (*2D contour color ramp shader*) or as a *Single color* with an associated *Mesh color*
- *Show arrows*: displays arrows on mesh layer dataset 3D entity, based on the same vector dataset group used in the *vector 2D rendering*. They are displayed using the 2D color setting. It's also possible to define the *Arrow spacing* and, if it's of a *Fixed size* or scaled on magnitude. This spacing setting defines also the max size of arrows because arrows can't overlap.

14.3.6 Rendering Properties



14.13 Fig. : Mesh rendering properties

Under the *Scale dependent visibility* group box, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scale, defining a range of scale in which mesh elements will be visible. Out of this range, they are hidden. The
Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See *Nuo mastelio priklausomo matomumo parinkimas* for more information.

Pastaba

You can also activate scale dependent visibility on a layer from within the *Layers* panel: right-click on the layer and in the contextual menu, select *Set Layer Scale Visibility*.

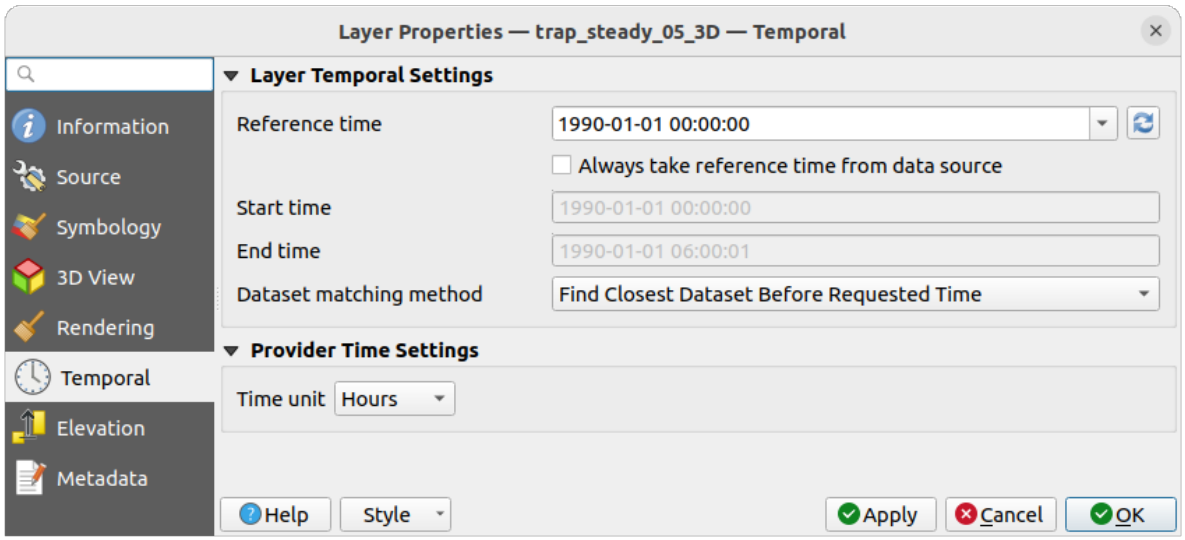
As mesh layers can have millions of faces, their rendering can sometimes be very slow, especially when all the faces are displayed in the view whereas they are too small to be viewed. To speed up the rendering, you can simplify the mesh layer, resulting in one or more meshes representing different *levels of detail* and select at which level of detail you would like QGIS to render the mesh layer. Note that the simplify mesh contains only triangular faces.

From the *Rendering* tab, check *Simplify mesh* and set:

- a *Reduction factor*: Controls generation of successive levels of simplified meshes. For example, if the base mesh has 5M faces, and the reduction factor is 10, the first simplified mesh will have approximately 500 000 faces, the second 50 000 faces, the third 5000,… If a higher reduction factor leads quickly to simpler meshes (i.e. with triangles of bigger size), it produces also fewer levels of detail.
- *Minimum triangle size*: the average size (in pixels) of the triangles that is permitted to display. If the average size of the mesh is lesser than this value, the rendering of a lower level of details mesh is triggered.

14.3.7 Laiko savybės

The *Temporal* tab provides options to control the rendering of the layer over time. It allows to dynamically display temporal values of the enabled dataset groups. Such a dynamic rendering requires the *temporal navigation* to be enabled over the map canvas.



14.14 Fig. : Mesh Temporal properties

Layer temporal settings

- *Reference time* of the dataset group, as an absolute date time. By default, QGIS parses the source layer and returns the first valid reference time in the layer's dataset group. If unavailable, the value will be set by the project time range or fall back to the current date. The *Start time* and *End time* to consider are then calculated based on the internal timestamp step of the dataset.

  It is possible to set a custom *Reference time* (and then the time range), and revert the changes using the Reload from provider button. With *Always take reference time from data source* checked, you ensure that the time properties are updated from the file each time the layer is reloaded or the project reopened.

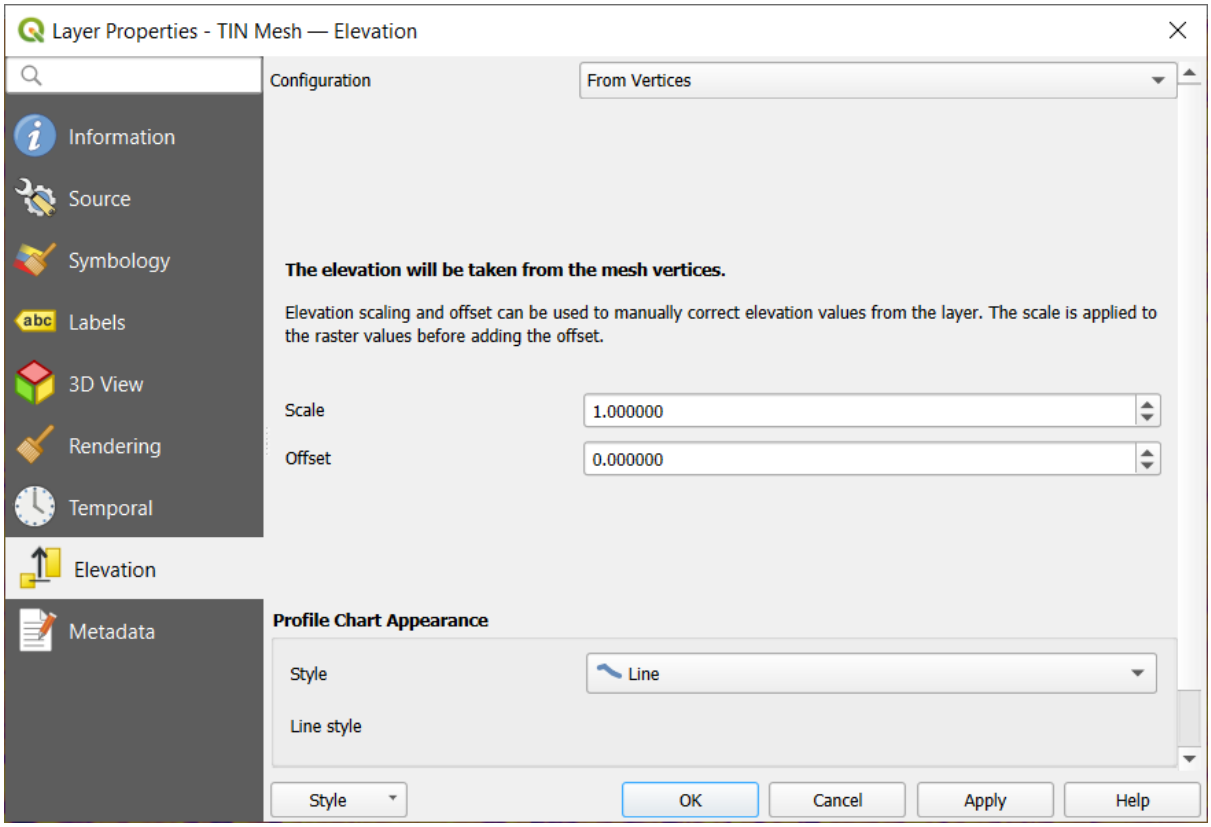
- *Dataset matching method*: determines the dataset to display at the given time. Options are *Find closest dataset before requested time* or *Find closest dataset from requested time (after or before)*.

Provider time settings

- *Time unit* extracted from the raw data, or user defined. This can be used to align the speed of the mesh layer with other layers in the project during map time navigation. Supported units are *Seconds*, *Minutes*, *Hours* and *Days*.

14.3.8 Aukščio savybės

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view* and *2D map view* and its appearance in the *profile tool charts*. Specifically, you can configure how heights from your dataset are interpreted:



14.15 Fig. : Mesh Elevation properties

- *From vertices*: how the mesh layer vertices Z values should be interpreted as terrain elevation. You can apply a *Scale* factor and an *Offset*. This setting is available for *3D map view* and *profile tool charts*.
- *Fixed Elevation Range*: the mesh layer is linked to a fixed elevation range. This mode is applicable when a layer has either a single fixed elevation or a range (slice) of elevation values. If a range is specified, mesh values will be extruded over this range. You can set the *Lower* and *Upper* elevation range values for the layer, and specify whether the lower or upper *Limits* are inclusive or exclusive. When enabled, the layer will only be visible in *elevation filtered 2D maps* when the layer's range is included in the map's Z range.
- *Fixed Elevation Range Per Group*: each group in the mesh layer is associated with a fixed elevation range. This mode can be used when a layer has elevation data exposed through different dataset groups. This feature is exposed as a user-editable table for dataset groups with lower and upper values. You can either populate the lower and upper values manually or use an ε *Expression* to auto-fill all group values based on an expression. When enabled, the layer will be filtered in the *2D map view*, displaying only values within the map filter ranges. In *3D map view* or *profile tool charts*, the full extent of the layer will be displayed, ignoring the filtering.
- *Profile Chart Appearance*: controls the rendering of the mesh elements elevation in the profile chart. The profile *Style* can be set as:
  - a *Line* with a specific *Line style*
  - an elevation surface rendered using a fill symbol either above (*Fill above*) or below (*Fill below*) the elevation curve line. The surface symbology is represented using:
    - a *Fill style*
    - and a *Limit*: the maximum (respectively minimum) altitude determining how high the fill surface will be

14.3.9 Metaduomenų savybės

The *Metadata* tab provides you with options to create and edit a metadata report on your layer. See *Metaduomenys* for more information.

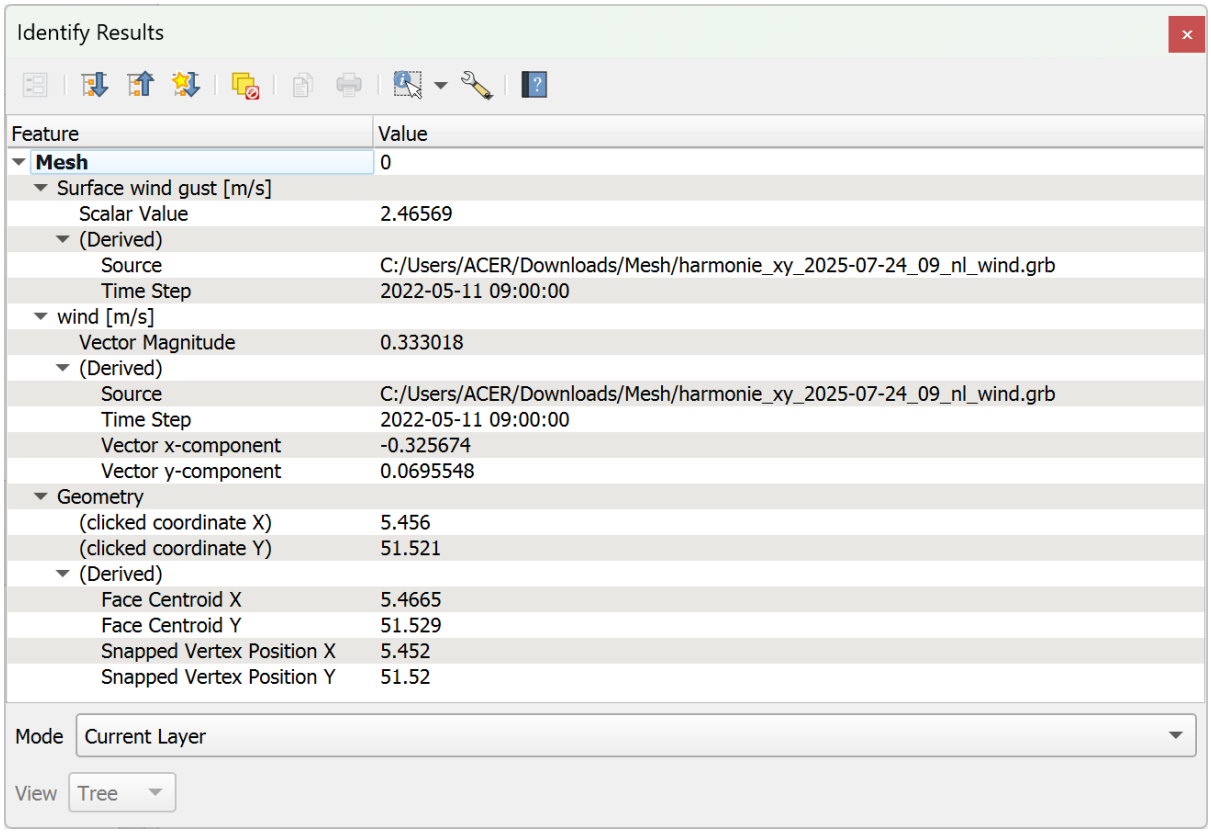
14.4 Exploring mesh datasets

To identify mesh elements and their associated datasets, use the *Identify Features* tool from the *Attributes* toolbar. Click on a mesh element in the map canvas and the *Identify Results* panel opens, displaying information about the mesh layer and the identified element. The panel shows:

- Layer name
- Active dataset groups (scalar and/or vector) for the clicked element will be displayed first. On the other hand, inactive dataset groups are shown afterwards. For each group, the following derived information may also be shown, if applicable:
  - **Source** of the dataset (the file where it is stored).
  - **Time step** of the dataset value being displayed. For non-temporal dataset groups, this field remains empty.
  - **Vector x-component** and **Vector y-component** values for vector datasets.

Please note that when temporal navigation is disabled, the results only contain values from the *static dataset groups* defined.

- Geometry information shows the coordinates of the clicked point and may contain derived information in the following:
  - Centroid of the face.
  - Coordinates of a snapped vertex (if the click was snapped to a vertex).
  - Center of a snapped edge (if the click was snapped to an edge).



14.16 Fig. : Mesh Identify Results panel

14.5 Editing a mesh layer

QGIS allows to *create a mesh layer* from scratch or based on an existing one. You can create/modify the geometries of the new layer whom you can assign datasets afterwards. It's also possible to edit an existing mesh layer. Because the editing operation requires a frames-only layer, you will be asked to either remove any associated datasets first (make sure you have them available if they still are necessary) or create a copy (only geometries) of the layer.

Pastaba

QGIS does not allow to digitize edges on mesh layers. Only vertices and faces are mesh elements that can be created. Also not all supported mesh formats can be edited in QGIS (see permissions).

14.5.1 Overview of the mesh digitizing tools

To interact with or edit a base mesh layer element, following tools are available.

14.4 Table : Tools for mesh digitizing

Label	Tikslas	Location
Current Edits	Access to save, rollback or cancel changes in all or selected layers simultaneously	*Digitizing* toolbar
Toggle to Edit	Turn on/off the layer in edit mode	*Digitizing* toolbar
Save Edits	Save changes done to the layer	*Digitizing* toolbar
Undo	Undo the last change(s) - `Ctrl+Z`	*Digitizing* toolbar
Redo	Redo the last undone action(s) - `Ctrl+Shift+Z`	*Digitizing* toolbar
Enable Advanced Digitizing tools	Turn on/off the *Advanced Digitizing Panel*	*Advanced Digitizing* toolbar
Reindex Faces and Vertices	Recreate index and renumber the mesh elements for optimization	*Mesh* menu
Digitize Mesh Elements	Select/Create vertices and faces	*Mesh* *Digitizing* toolbar
Select Mesh Elements by Polygon	Select vertices and faces overlapped by a drawn polygon	*Mesh* *Digitizing* toolbar
Select Mesh Elements by Expressio	Select vertices and faces using an expression	*Mesh* *Digitizing* toolbar
Transform Vertices Coordinates	Modify coordinates of a selection of vertices	*Mesh* *Digitizing* toolbar
Force by Selected Geometries	Split faces and constrain Z value using a linear geometry	*Mesh* *Digitizing* toolbar

14.5.2 Exploring the Z value assignment logic

When a mesh layer is turned into edit mode, a *Vertex Z value* widget opens at the top right of the map canvas. By default, its value corresponds to the *Default Z value* set in *Settings* ► *Options* ► *Digitizing* tab. When there are selected vertices, the widget displays the average Z value of the selected vertices.

During editing, the *Vertex Z value* is assigned to new vertices. It is also possible to set a custom value: edit the widget, press `Enter` and you will override the default value and make use of this new value in the digitizing process. Click the icon in the widget to reset its value to the Options default value.

Rules of assignment

This is configured in the drop-down menu of the *Digitize mesh elements* tool as *New vertex Z value* option. Following methods are available:

14.5 Table : Z value assignment methods

Method	Description
Project terrain	Z value is always taken or calculated from the *project reference terrain*. • If it is **Flat terrain**, the value corresponds to the *Terrain height* set in the project properties. • If it is a **Mesh** layer, the value is interpolated from the surrounding vertices of the mesh. • If it is a **DEM (raster layer)**, the value is assigned from the raster pixel value. • If there is no terrain under the new vertex, the value falls back to the *Default Z value* set in *digitizing options*.
Z widget	Z value is taken from the *Vertex Z value* widget or the *Advanced Digitizing Panel* *Z* widget (if it is in Locked state).
Prefer mesh, then Z widget	Interpolates the Z value from the active mesh layer if the vertex is on an edge or face, otherwise falls back to the *Vertex Z value* or *Advanced Digitizing Panel* *Z* widget (if it is in Locked state).
Prefer mesh, then terrain	Interpolates the Z value from the active mesh layer if the vertex is on the edge or face, otherwise uses the project reference terrain value.

The following detailed logic describes the behavior of the **Prefer mesh, then Z widget** strategy.

14.6 Table : Matrix of Z value assignment to new vertex

Vertex creation	Vertices selected?	Source of assigned value	Assigned Z Value
„Free“ vertex, not connected to any face or edge of a face	Ne	*Vertex Z value*	Default or user defined
		Advanced Digitizing Panel (if *z* widget is in Locked state)	*z* widget if in Locked state
	Taip	*Vertex Z value*	Average of the selected vertices or user-defined
Vertex on an edge	—	Mesh layer	Interpolated from the edge’s vertices
Vertex on a face	—	Mesh layer	Interpolated from the face’s vertices
Vertex snapped to a 2D vector feature	—	*Vertex Z value*	Default or user defined
Vertex snapped to a 3D vector vertex	—	Vector layer	Vertex
Vertex snapped to a 3D vector segment	—	Vector layer	Interpolated along the vector segment

Pastaba

The *Vertex Z value* widget is deactivated if the *Advanced Digitizing Panel* is enabled and no mesh element is selected. The latter’s *z* widget then rules the Z value assignment.

Modifying Z value of existing vertices

To modify the Z value of vertices, the most straightforward way is:

1. Select one or many vertices. The *Vertex Z value* widget will display the average height of the selection.
2. Change the value in the widget.
3. Press `Enter`. The entered value is assigned to the vertices and becomes the default value of next vertices.

Another way to change the Z value of a vertex is to move and snap it on a vector layer feature with the Z value capability. If more than one vertex are selected, the Z value can't be changed in this way.

The *Transform mesh vertices* dialog also provides means to modify the Z value of a selection of vertices (along with their X or Y coordinates).

14.5.3 Selecting mesh elements

To select mesh elements, you can use the following tools:

- Digitize Mesh Elements to select different mesh elements, see more at *Using Digitize Mesh Elements*.
- Select Mesh Elements by Polygon to select different mesh elements by polygon, see more at *Using Select Mesh Elements by Polygon*.
- Select Mesh Elements by Expression to select different mesh elements by expression, you can choose to *Select by vertices* or *Select by faces*, see more at *Using Select Mesh Elements by Expression*.
- Select Isolated Vertices to select all vertices that are not part of any mesh face.
- Select All Vertices to select all vertices of the mesh layer.

Using *Digitize Mesh Elements*

Activate the Digitize Mesh Elements tool. Hover over an element and it gets highlighted, allowing you to select it.

- Click on a vertex, and it is selected.
- Click on the small square at the center of a face or an edge, and it gets selected. Connected vertices are also selected. Conversely, selecting all the vertices of an edge or a face also selects that element.
- Drag a rectangle to select overlapping elements (a selected face comes with all their vertices). Press `Alt` key if you want to select only completely contained elements.
- To add elements to a selection, press `Shift` while selecting them.
- To remove an element from the selection, press `Ctrl` and reselect it. A deselected face will also deselect all their vertices.

Using *Select Mesh Elements by Polygon*

Activate the Select Mesh Elements by Polygon tool and:

- Draw a polygon (left-click to add vertex, `Backspace` to undo last vertex, `Esc` to abort the polygon and right-click to validate it) over the mesh geometries. Any partially overlapping vertices and faces will get selected. Press `Alt` key while drawing if you want to select only completely contained elements.
- Right-click over the geometry of a vector layer's feature, select it in the list that pops up and any partially overlapping vertices and faces of the mesh layer will get selected. Use `Alt` while drawing to select only completely contained elements.

- To add elements to a selection, press `Shift` while selecting them.
- To remove an element from the selection, press `Ctrl` while drawing over the selection polygon.

Using *Select Mesh Elements by Expression*

Another tool for mesh elements selection is Select Mesh Elements by Expression. When pressed, the tool opens the mesh *expression selector dialog* from which you can:

1. Select the method of selection:
   - *Select by vertices*: applies the entered expression to vertices, and returns matching ones and their eventually associated edges/faces
   - *Select by faces*: applies the entered expression to faces, and returns matching ones and their associated edges/vertices
2. Write the expression of selection. Depending on the selected method, available functions in the *Meshes group* will be filtered accordingly.
3. Run the query by setting how the selection should behave and pressing:
   - *Select*: replaces any existing selection in the layer
   - *Add to current selection*
   - *Remove from current selection*

14.5.4 Modifying mesh elements

Adding vertices

To add vertices to a mesh layer:

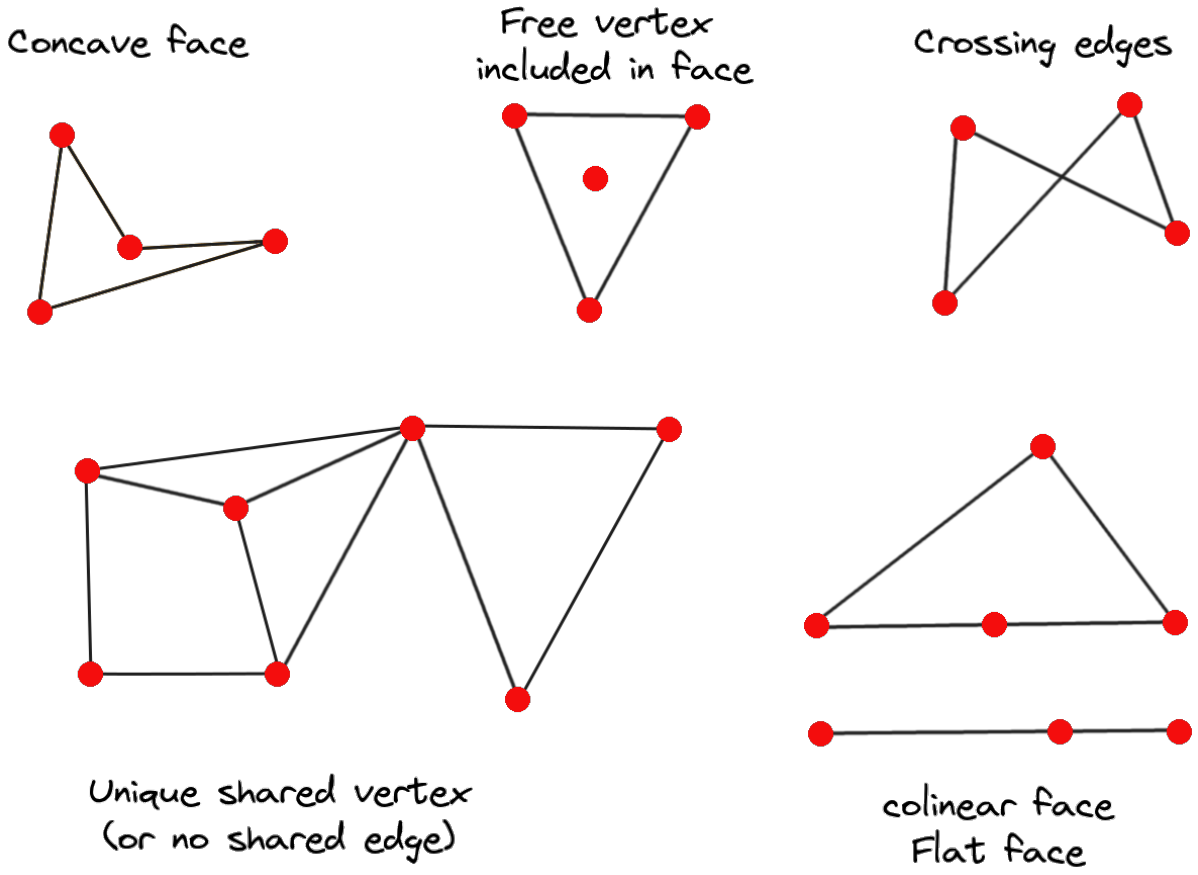
1. Press the Digitize mesh elements button
2. A *Vertex Z value* widget is always available on the top right corner of the map canvas during digitizing. It allows setting or overriding the Z coordinate assigned to new vertices, depending on the selected *Z value assignment method*.
3. You can choose other method for assigning Z values to new vertices from the drop-down menu in the Digitize mesh elements tool.
4. Then double-click:
   - outside a face: adds a „free vertex“, that is a vertex not linked to any face. This vertex is represented by a red dot when the layer is in editing mode.
   - on the edge of existing face(s): adds a vertex on the edge, splits the touching face(s) into triangles connected to the new vertex.
   - inside a face: splits the face into triangles whose edges connect the surrounding vertices to the new vertex.

     With the *Refine neighboring faces when adding vertices* option enabled in the Digitize mesh elements button drop-down menu, a check is applied to triangular faces that share at least one vertex with the face the new vertex is added to. If their edges do not satisfy Delaunay triangulation rules, they are flipped accordingly.

Adding faces

To add faces to a mesh layer:

1. Press the Digitize mesh elements button
2. A *Vertex Z value* widget appears on the top right corner of the map canvas. Set this value to the Z coordinate you would like to assign to the subsequent vertices.
3. Hover over a vertex and click the small triangle that appears next to it.
4. Move the cursor to the next vertex position; you can snap to existing vertex or left-click to add a new one.
5. Proceed as above to add as many vertices as you wish for the face. Press `Backspace` button to undo the last vertex.
6. While moving the mouse, a rubberband showing the shape of the face is displayed. If it is shown in green, then the expected face is valid and you can right-click to add it to the mesh. If it is red, the face is not valid (e.g. because it self-intersects, overlaps an existing face or vertex, creates a hole, …) and can't be added. You'd need to undo some vertices and fix the geometry.
7. Press `Esc` to abort the face digitizing.
8. Right-click to validate the face.



14.17 Fig. : Examples of invalid mesh

Removing mesh elements

1. *Select the target elements*
2. Enable the Digitize mesh elements tool
3. Right-click and select:
   - *Remove Selected Vertices and Fill Hole(s)* or press `Ctrl+Del`: removes vertices and linked faces and fills the hole(s) by triangulating from the neighbor vertices
   - *Remove Selected Vertices Without Filling Hole(s)* or press `Ctrl+Shift+Del`: removes vertices and linked faces and do not fill hole(s)
   - *Remove Selected Face(s)* or press `Shift+Del`: removes faces but keeps the vertices

These options are also accessible from the contextual menu when hovering over a single item without selecting.

Moving mesh elements

To move vertices and faces of a mesh layer:

1. *Select the target elements*
2. Enable the Digitize mesh elements tool
3. To start moving the element, click on a vertex or the centroid of a face/edge
4. Move the cursor to the target location (snapping to vector features is supported).
5. If the new location does not generate an *invalid mesh*, the moved elements appear in green. Click again to release them at this location. Faces whose vertices are all selected are translated, their neighbors are reshaped accordingly.

Transforming mesh vertices

The Transform Vertices Coordinates tool gives a more advanced way to move vertices, by editing their X, Y and/or Z coordinates thanks to expressions.

1. Select the vertices you want to edit the coordinates
2. Press Transform Vertices Coordinates. A dialog opens with a mention of the number of selected vertices. You can still add or remove vertices from the selection.
3. Depending on the properties you want to modify, you need to check the *X coordinate*, *Y coordinate* and/or *Z value*.
4. Then enter the target position in the box, either as a numeric value or an expression (using the Expression dialog)
5. With the Import Coordinates of the Selected Vertex pressed, the X, Y and Z boxes are automatically filled with its coordinates whenever a single vertex is selected. You can adjust Z value by using *Get the Z value from the project terrain* button and in that case the Z box is deactivated. A convenient and quick way to adjust vertices individually.
6. Press *Preview Transform* to simulate the vertices new location and preview the mesh with transformation.
   - If the preview is green, transformed mesh is valid and you can apply the transformation.
   - If the preview is red, the transformed mesh is invalid and you can not apply the transformation until it is corrected.
7. Press *Apply Transform* to modify the selected coordinates for the set of vertices.

Reshaping mesh geometry

The contextual menu

1. Enable the Digitize mesh elements
2. Select mesh item(s), or not
3. Hover over a mesh element, it gets highlighted.
4. Right-click and you can:
   - *remove the item(s)*
   - *Split Selected Face(s)* (*Split Current Face*): splits the face you are hovering over or each selected quad mesh faces into two triangles
   - *Delaunay Triangulation with Selected vertices*: builds triangular faces using selected free vertices.
   - *Refine Selected Face(s)* (*Refine Current Face*): splits the face into four faces, based on vertices added at the middle of each edge (a triangle results into triangles, a quad into quads). Also triangulates adjacent faces connected to the new vertices.

The edge markers

When the Digitize mesh elements is active and you hover over an edge, the edge is highlighted and it is possible to interact with it. Depending on the context, following markers may be available:

- a **square**, at the center of the edge: click on it to select extremity vertices.
- a **cross** if the two faces on either side can be merged: click on it to delete the edge and merge the faces.
- a **circle** if the edge is between two triangles: Click on it to flip the edge, i.e. connect it instead to the two other „free“ vertices of the faces

The *Force by Selected Geometries* tool

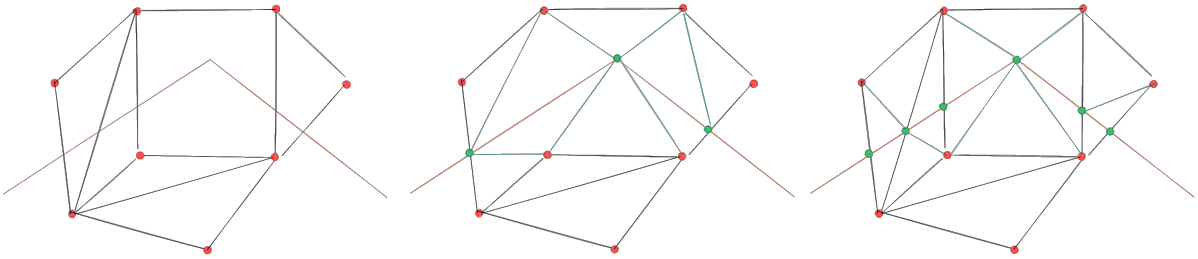
The Force by Selected Geometries tool provides advanced ways to apply break lines using lines geometry. A break line will force the mesh to have edges along the line. Note that the break line will not be considered persistent once the operation is done; resulting edges will not act as constraints anymore and can be modified like any other edge. This can be used for example to locally modify a mesh layer with accurate lines, as river banks or border of road embankments.

1. Enable the Force by Selected Geometries tool
2. Indicate the geometry to use as „forcing line“; it can be:
   - picked from a line or polygon feature in the map canvas: right-click over the vector feature and select it from the list in the contextual menu.
   - a virtual line drawn over the mesh frame: left-click to add vertices, right-click for validation. Vertices Z value is set through the *Vertex Z value* widget or the *z* widget if the *Advanced Digitizing Panel* is on. If the line is snapped to a mesh vertex or a 3D vector feature's vertex or segment, the new vertex takes the snapped element Z value.

Mesh faces that overlap the line geometry or the polygon's boundary will be affected in a way that depends on options you can set from the Force by Selected Geometries tool drop-down menu:

- *Add new vertex on intersecting edges*: with this option, a new vertex is added each time the forcing line intersect an edge. This option leads to split along the line each encountered faces.

Without this option, encountered faces are removed and replaced by faces coming from a triangulation with only the existing vertices plus the vertices of the forcing lines (new vertices are also added on the boundary edge intersecting the forcing lines).



14.18 Fig. : Force Mesh using a line geometry - Results without (middle) and with (right) new vertex on edges intersection

- *Interpolate Z value from*: set how the new vertices Z value is calculated. It can be from:
  - the *Mesh* itself: the new vertices Z value is interpolated from vertices of the face they fall within
  - or the *Forcing line*: if the line is defined by a 3D vector feature or a drawn line then the new vertices Z value is derived from its geometry. In case of 2D line feature, the new vertices Z value is the *Vertex Z value*.
- *Tolerance*: when an existing mesh vertex is closer to the line than the tolerance value, do not create new vertex on the line but use the existing vertex instead. The value can be set in *Meters at Scale* or in *Map Units* (more details at *Vienetų parinkiklis*).

14.5.5 Reindexing meshes

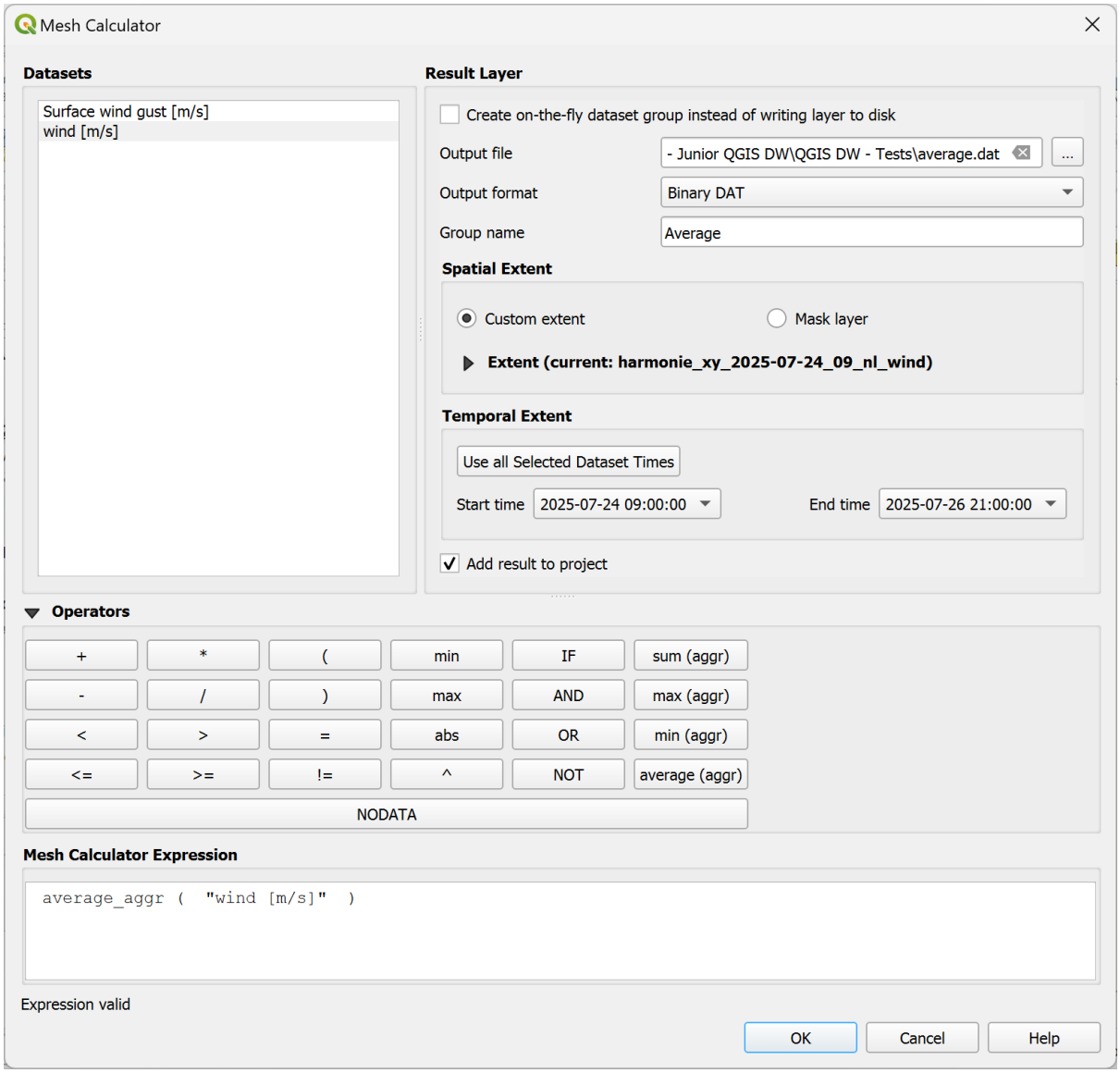
During edit, and in order to allow quick undo/redo operations, QGIS keeps empty places for deleted elements, which may lead to growing memory use and inefficient mesh structuring. The *Mesh* ► *Reindex Faces and Vertices* tool is designed to remove these holes and renumber the indices of faces and vertices so that they are continuous and somewhat reasonably ordered. This optimizes relation between faces and vertices and increases the efficiency of calculation.

Pastaba

The *Reindex Faces and Vertices* tool saves the layer and clear the undo/redo stacks, disabling any rollback.

14.6 Mesh Calculator

The *Mesh Calculator* tool from the top *Mesh* menu allows you to perform arithmetic and logical calculations on existing dataset groups to generate a new dataset group (see Fig. 14.19).



14.19 Fig. : Mesh Calculator

The *Datasets* list contains all dataset groups in the active mesh layer. To use a dataset group in an expression, double click its name in the list and it will be added to the *Mesh calculator expression* field. You can then use the operators to construct calculation expressions, or you can just type them into the box.

The *Result Layer* helps you configure properties of the output layer:

- *Create on-the-fly dataset group instead of writing layer to disk*:
  - If unchecked, the output is stored on disk as a new plain file. An *Output File* path and an *Output Format* are required.
  - If checked, a new dataset group will be added to the mesh layer. Values of the dataset group are not stored in memory but each dataset is calculated when needed with the formula entered in the mesh calculator. That virtual dataset group is saved with the project, and if needed, it can be removed or made persistent in file from the layer *Source* properties tab.

  In either case, you should provide a *Group Name* for the output dataset group.
- The *Spatial extent* to consider for calculation can be:
  - defined as a *Custom extent* using the *extent selector* widget.

  - defined by a polygon layer (*Mask layer*) of the project: the polygon features geometry are used to clip the mesh layer datasets

- The *Temporal extent* to take into account for datasets can be set with the *Start time* and *End time* options, selected from the existing dataset groups timesteps. They can also be filled using the *Use all selected dataset times* button to take the whole range.

- With the *Add result to project* checkbox, the result layer will automatically be added to the legend area and can be visualized. Checked by default.

The *Operators* section contains all available operators. To add an operator to the mesh calculator expression box, click the appropriate button. Mathematical calculations (`+`, `-`, `*`, ...) and statistical functions (`min`, `max`, `sum (aggr)`, `average (aggr)`, ...) are available. Conditional expressions (`=`, `!=`, `<`, `>=`, `IF`, `AND`, `NOT`, ...) return either 0 for false and 1 for true, and therefore can be used with other operators and functions. The `NODATA` value can also be used in the expressions.

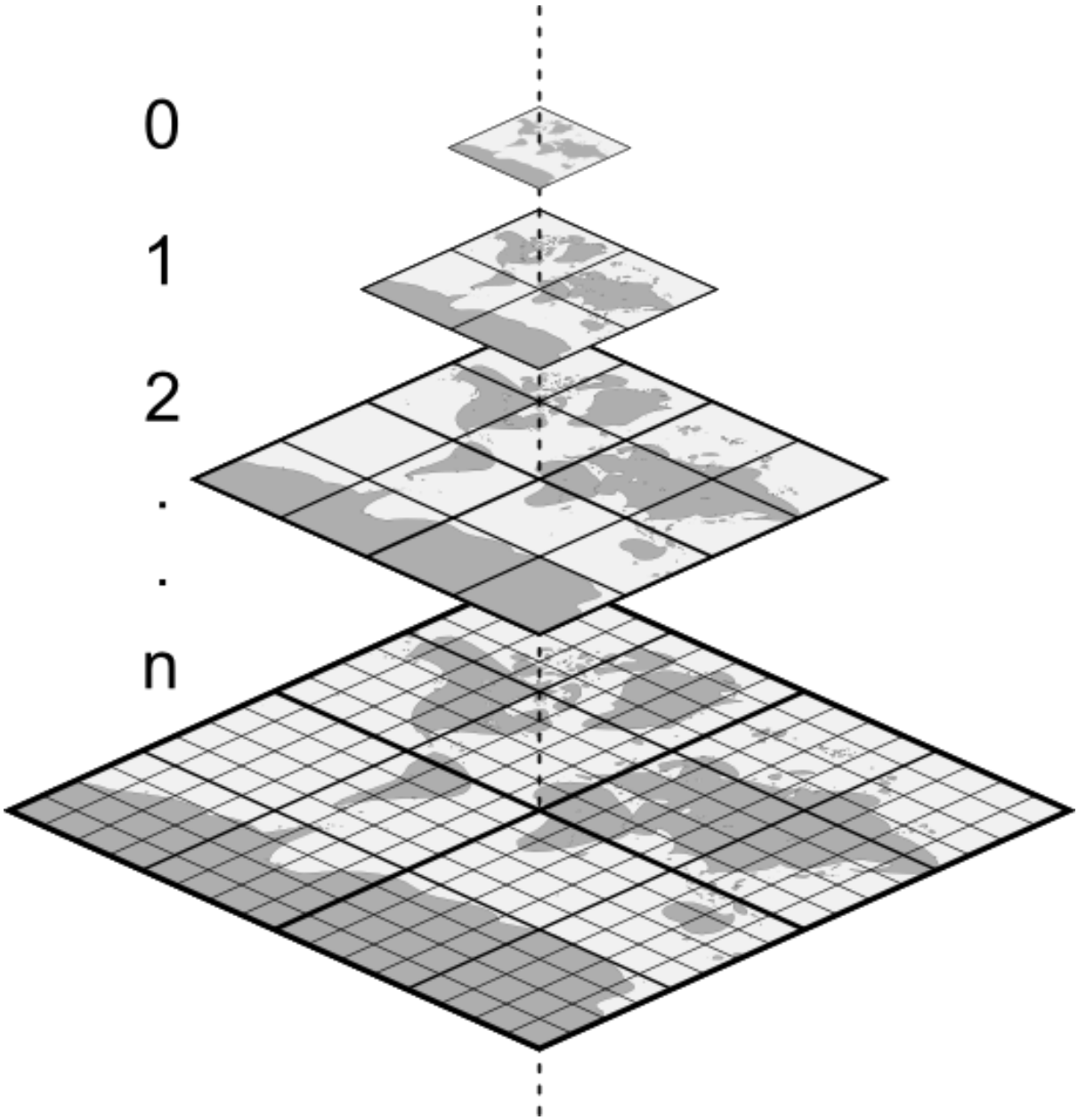
The *Mesh Calculator Expression* widget shows and lets you edit the expression to execute.

SKYRIUS 15

Working with Vector Tiles

15.1 What are Vector Tiles?

Vector tiles are packets of geographic data, packaged into pre-defined roughly-square shaped „tiles“ for transfer over the web. They combine pre-rendered raster map tiles and vector map tiles. The vector tile server returns vector map data, which has been clipped to the boundaries of each tile, instead of a pre-rendered map image. The clipped tiles represent the zoom-levels of the vector tile service, derived from a pyramid approach. Using this structure, the data-transfer is reduced in comparison to un-tiled vector maps. Only data within the current map view, and at the current zoom level need to be transferred. Also, compared to a tiled raster map, data transfer is also greatly reduced, as vector data is typically much smaller than a rendered bitmap. Vector tiles do not have any styling information assigned so QGIS needs to apply a cartographic style in order to display the data.



15.1 Fig. : Pyramid structure of vector tiles with zoom-levels

15.2 Supported Formats

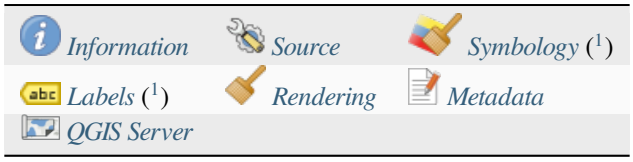
There is support for vector tiles through:

- remote sources (HTTP/S) - with XYZ template - `type=xyz&url=http://example.com/{z}/{x}/{y}.pbf`
- local files - with XYZ template - e.g. `type=xyz&url=file:///path/to/tiles/{z}/{x}/{y}.pbf`
- local MBTiles database - e.g. `type=mbtiles&url=file:///path/to/file.mbtiles`

To load a vector tiles dataset into QGIS, use the *Vector Tile* tab in the *Data Source Manager* dialog. Read *Vektorinių kaladėlių paslaugų naudojimas* for more details.

15.3 Vector Tiles Dataset Properties

The vector tiles *Layer Properties* dialog provides the following sections:



15.3.1 Informacinės savybės

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata on the current layer. Provided information are:

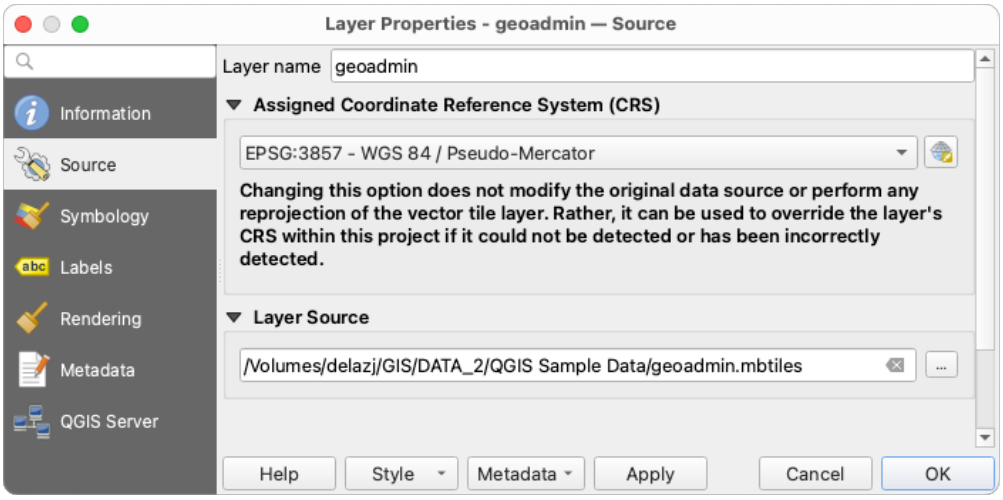
- based on the provider of the layer: name, URI, source type and path, number of zoom levels
- custom properties, used to store in the active project additional information about the layer. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- the Coordinate Reference System: name, units, method, accuracy, reference (i.e., whether it is static or dynamic)
- picked from the *filled metadata*: access, extents, links, contacts, history…

15.3.2 Source Properties

The *Source* tab displays basic information about the selected vector tile, including:

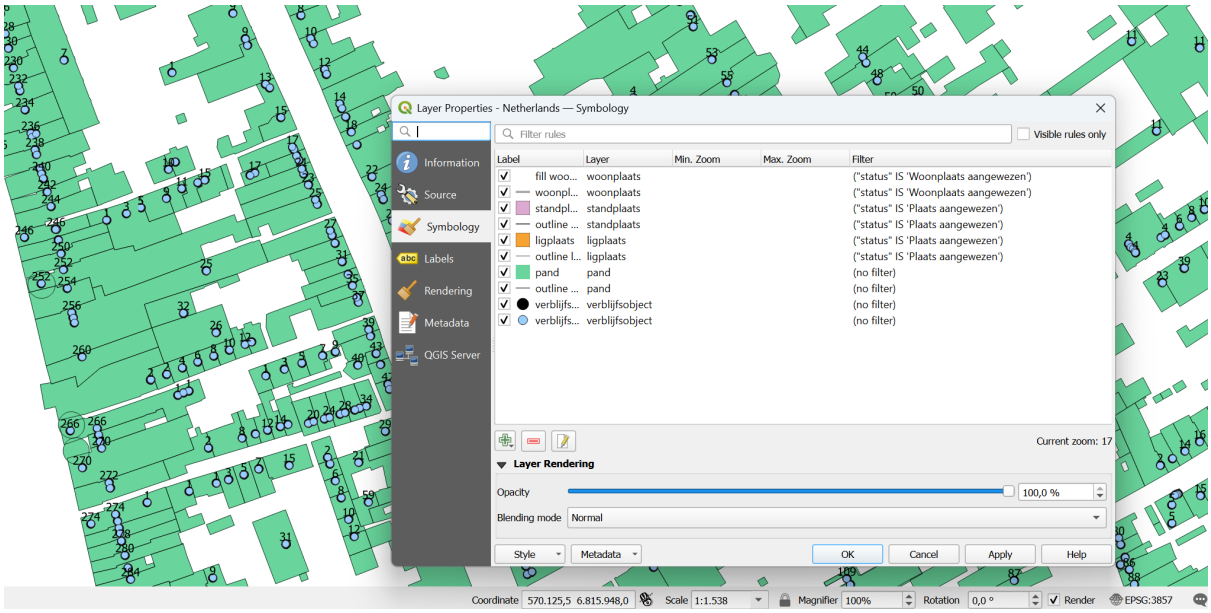
- the *Layer name* to display in the *Layers Panel*;
- the *Coordinate Reference System*: Displays the layer's *Coordinate Reference System (CRS)*. You can change the layer's CRS, by selecting a recently used one in the drop-down list or clicking on the Select CRS button (see *Koordinačių atskaitos sistemos parinkiklis*). Use this process only if the layer CRS is wrong or not specified.
- For file based vector tiles (`.vtpk` or `.mbtiles`), a *Layer source* group indicates the path to the source of the dataset and allows for replacing the loaded layer. Edit the path shown in the text box or press … Browse to select another file on the disk.

[1] Also available in the *Layer styling panel*



15.2 Fig. : Vector Tiles Properties - Source Dialog

15.3.3 Symbology and Labels Properties



15.3 Fig. : Vector Tile Layer Symbology

Setting rules

As vector tiles consist of point, line and polygon geometries, the respective symbols are available. To apply a cartographic style (with symbology and/or labels), you can either:

- Use a *Style URL* when creating the *Vector Tiles Connection*. The symbology will be shown immediately in the *Symbology* tab after the layer is loaded in QGIS.
- Or build your own symbology and labeling in the corresponding tabs of the layer properties. By default, QGIS assigns an identical symbol to the features based on their geometry type.

In both cases, setting a style for a vector tile relies on a set of *rules* applied to the features, indicating:

- a vector *symbol* or *label*

- a *Label*, a title for comprehensive identification of the rule
- the name of a particular *Layer* the rule should apply to, if not applied to `(all layers)`
- a *Min. Zoom* and a *Max. Zoom*, for the range of display. Symbology and labeling can be dependent on the zoom level.
- a *Filter*, a QGIS expression to identify the features to apply the style to

Each rule is added pressing the Add rule button and selecting the type of symbols (*Marker*, *Line*, *Fill*) corresponding to the features geometry type. You can as well Remove selected rules or Edit current rule.

At the bottom the *Current Zoom* is shown. Check the *Visible rules only* option at the top of the dialog to filter the list of rules to only those that are visible at the current zoom level. This makes it easier to work with complex vector styling and to locate troublesome rules. The *Filter rules* text box also helps you easily find a rule, by searching the *Label*, *Layer* and *Filter* fields.

In Fig. 15.3 we set up style for the OpenStreetMap `landuse` layer. For better visibility most of the rules are deselected.

Layer rendering

From the *Symbology* tab, you can also set some options that invariably act on all features of the layer:

- *Nepermatomumas*: Naudodami šį įrankį jūs galite žemiau esantį sluoksnį matomu. Naudokite slankiklį, kad pritaikytumėte jūsų vektorinio sluoksnio matomumą pagal poreikius. Jūs taipogi galite tiksliai nurodyti matomumo procentą laukelyje greta slankiklio.
- *Blending mode*: You can achieve special rendering effects with these tools that you may previously only know from graphics programs. The pixels of your overlaying and underlaying layers are mixed through the settings described in *Suliejimo režimai*.

Stiliai

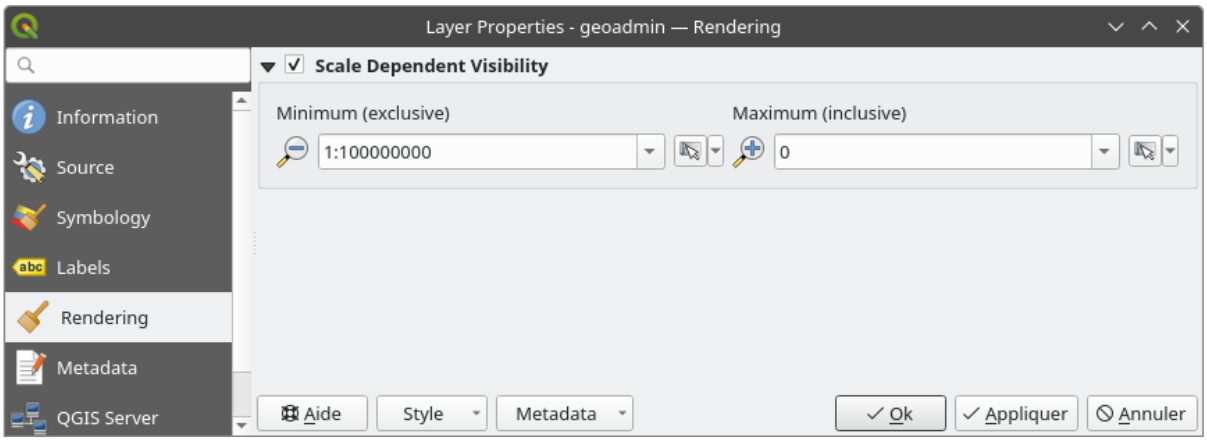
Available at the bottom of most of the tabs, the *Styles* ► menu provides shortcuts to save, load, create, switch styles to apply to the vector tiles. Vector tiles can have their style saved from QGIS as *QML* files and they can be imported as:

- *QML* files (*QML - The QGIS Style File Format*)
- *MapBox GL Json* style configuration files

More details at *Sluoksnio savybių įrašymas ir bendrinimas*.

15.3.4 Rendering Properties

Under *Scale dependent visibility*, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scales, defining a range of scales in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See *Nuo mastelio priklausomo matomumo parinkimas* for more information.



15.4 Fig. : Vector Tiles Properties - Rendering Dialog

15.3.5 Metaduomenų savybės

The *Metadata* tab provides you with options to create and edit a metadata report on your layer. See *Metaduomenys* for more information.

15.4 QGIS Server Properties

The *QGIS Server* tab helps you configure settings of the data when published by QGIS Server. The configuration concerns:

- *Description*: provides information to describe the data, such as *Short name*, *Title*, *Abstract*, a *Keyword list*, and a *Data URL* whose *Format* can be in `text/html`, `text/plain` or `application/pdf`.
- *Attribution*: a *Title* and *URL* to identify who provides the data
- *Metadata URL*: a list of *URL* for the metadata that can be of `FGDC` or `TC211` *Type*, and in `text/plain` or `text/xml` *Format*
- *Legend URL*: a *URL* for the legend, in either `image/png` or `image/jpeg` *Format*

SKYRIUS 16

Darbas su taškų debesimis

16.1 Įvadas į taškų debesis

Kas yra taškų debesis?

Taškų debesis yra trimatis erdvinis piešinys, sudarytas iš daug individualių duomenų taškų (iki milijardų ar trilijonų). Kiekvienas taškas turi x, y ir koordinatę. Priklausomai nuo to, kaip surinkta informacija, taškų debesys paprastai turi ir papildomus atributus, tokius kaip spalvų reikšmės ar intensyvumas. Šiuos atributus galima naudoti, pavyzdžiui, rodant taškų debesis skirtingomis spalvomis. QGIS taškų debesį galima naudoti kuriant trijų matmenų kraštovaizdžio (ar bet kurios kitos erdvės) piešinį.

Supported Formats

QGIS supports the data formats Entwine Point Tile (EPT) and LAS/LAZ. To work with point clouds, QGIS always saves the data in EPT. EPT is a storage format that consists of several files stored in a common folder. To allow quick access to the data, EPT uses indexing. For more information on the EPT format, see entwine homepage

If the data is in LAS or LAZ format, QGIS will convert it to EPT when it is loaded for the first time. Depending on the size of the file, this may take some time. In this process, a subfolder is created in the folder in which the LAS/LAZ file is located according to the scheme `ept_` + `name_LAS/LAZ_file`. If such a subfolder already exists, QGIS loads the EPT immediately (which leads to a reduced loading time).

Worth Knowing

In QGIS it is not (yet) possible to edit point clouds. If you want to manipulate your point cloud, you can use CloudCompare, an open source point cloud processing tool. Also the Point Data Abstraction Library (PDAL - similar to GDAL) offers you options to edit point clouds (PDAL is command line only).

Due to the large number of data points, it is not possible to display an attribute table of point clouds in QGIS. However, the *Identify tool* supports point clouds, so you can display all attributes, even of a single data point.

If you want to create a new layer, in the same or another supported format, from an existing point cloud layer, see *Creating new layers from an existing layer*.

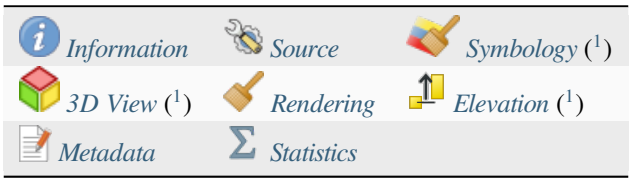
16.2 Point Clouds Properties

The *Layer Properties* dialog for a point cloud layer offers general settings for the layer and its rendering. It also provides information about the layer.

Norėdami pasiekti *Sluoksnio savybių* dialogą:

- In the *Layers* panel, double-click the layer or right-click and select *Properties…* from the context menu;
- Parinkus sluoksnį eikite į meniu *Sluoksniai* ► *Sluoksnio savybės…*.

The point cloud *Layer Properties* dialog provides the following sections:



Pastaba

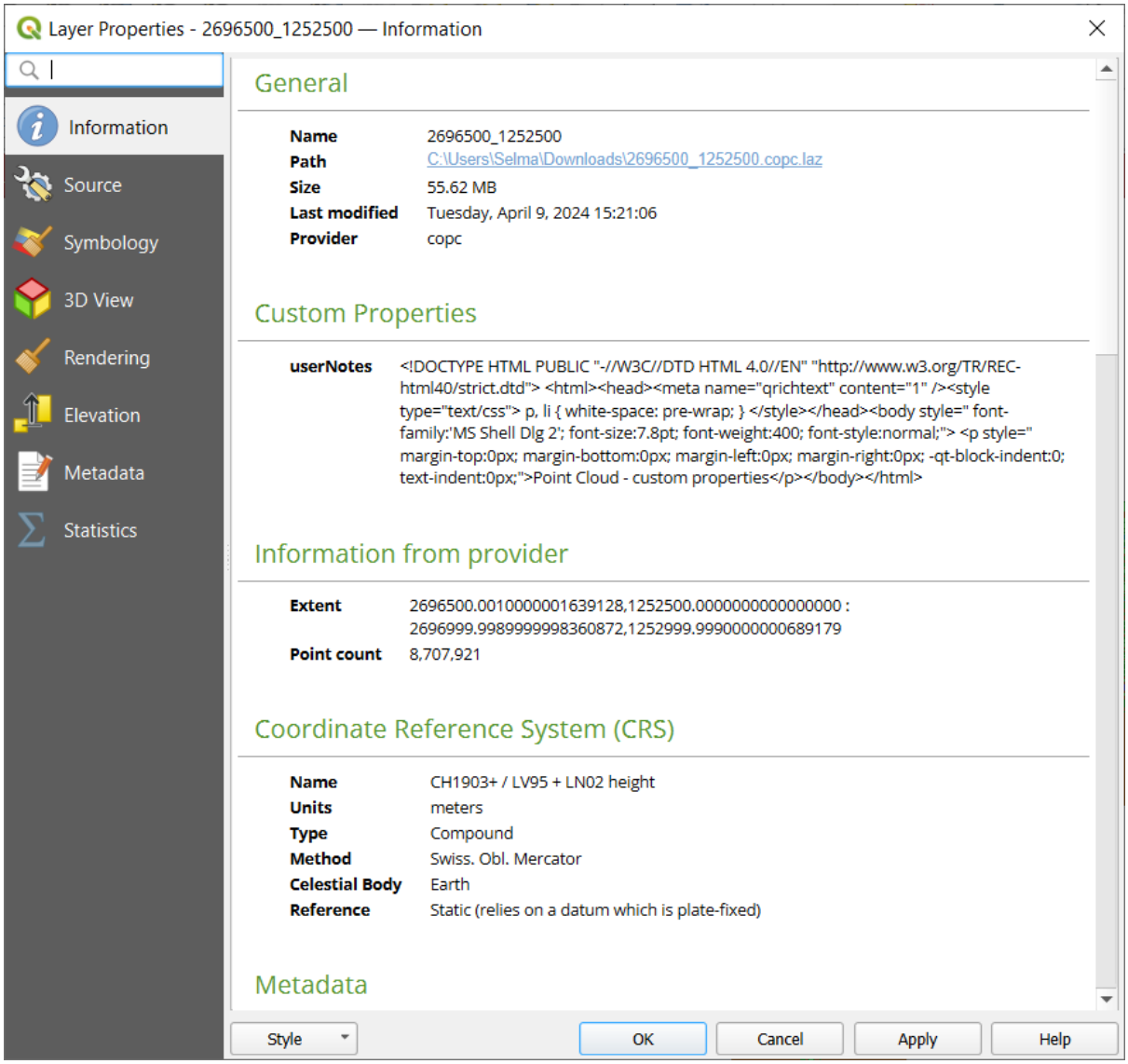
Most of the properties of a point cloud layer can be saved to or loaded from a `.qml` file using the *Style* menu at the bottom of the properties dialog. More details at *Sluoksnio savybių įrašymas ir bendrinimas*

16.2.1 Informacinės savybės

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata on the current layer. Provided information are:

- General such as name in the project, source path, last save time and size, the used provider
- custom properties, used to store in the active project additional information about the layer. Default custom properties may include *layer notes*. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- Based on the provider of the layer: extent and number of points
- The Coordinate Reference System: name, units, method, accuracy, reference (i.e. whether it's static or dynamic)
- Metadata delivered by the provider: creation date, version, data format, scale X/Y/Z, …
- Picked from the *Metadata* tab (where they can be edited): access, extents, links, contacts, history…

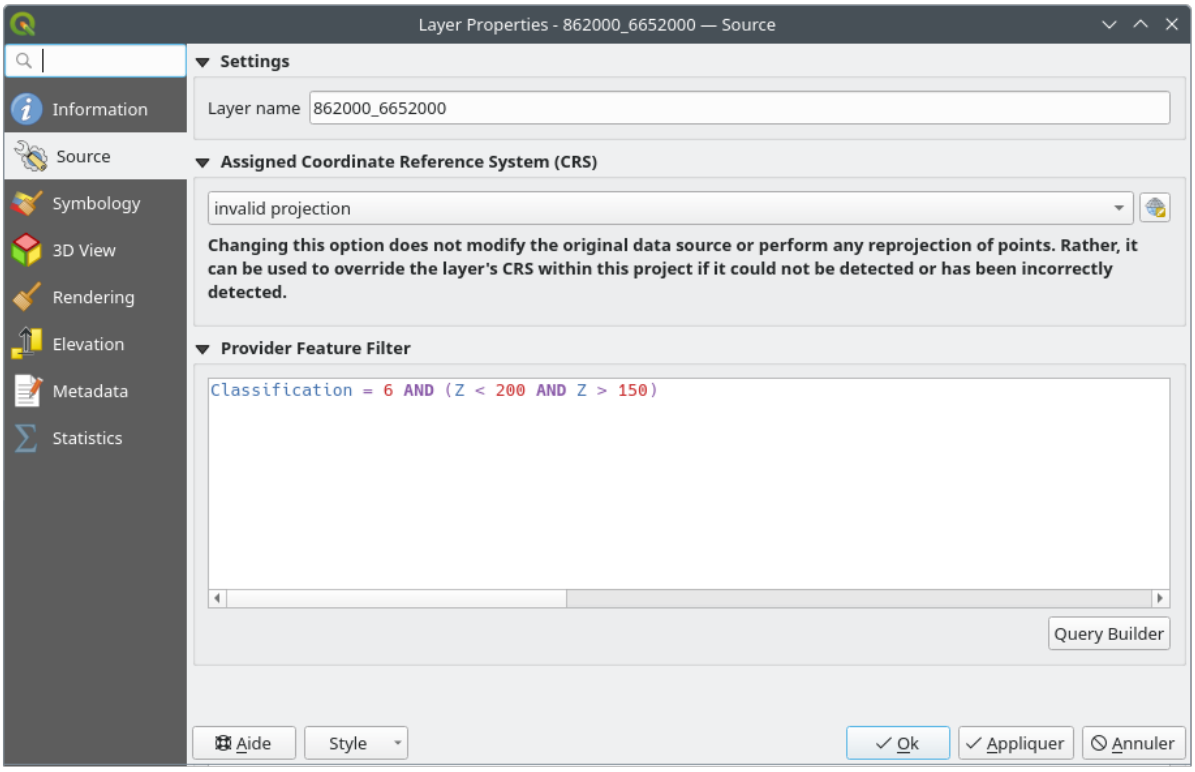
[1] Also available in the *Layer styling panel*



16.1 Fig. : Point cloud information tab

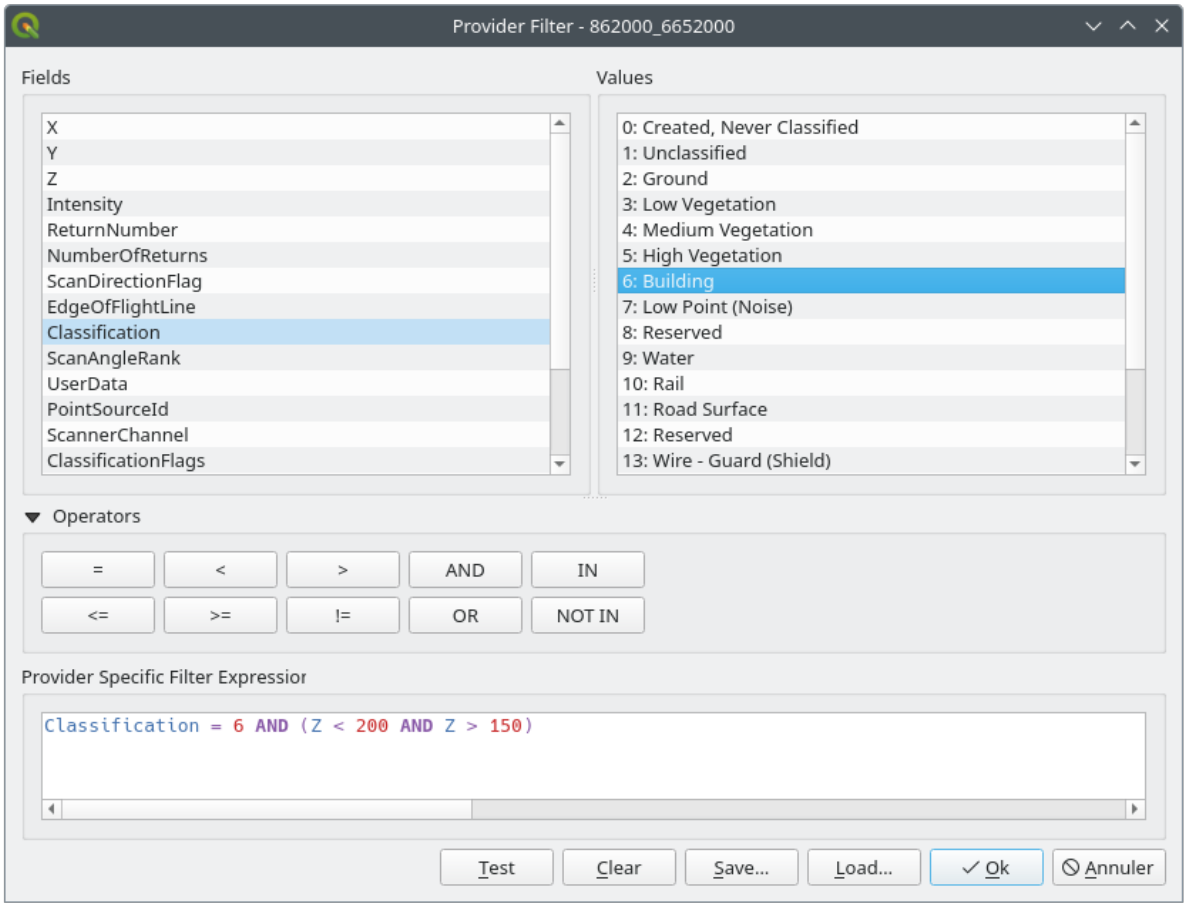
16.2.2 Source Properties

In the *Source* tab you can see and edit basic information about the point cloud layer:



16.2 Fig. : Point cloud source tab

- *Settings*: Set a Layer name different from the layer filename that will be used to identify the layer in the project (in the Layers Panel, with expressions, in print layout legend,…)
- *Assigned Coordinate Reference System (CRS)*: Here you can change the layer's assigned *Coordinate Reference System*, selecting a recently used one in the drop-down list or clicking on set Projection Select CRS button (see *Koordinačių atskaitos sistemos parinkiklis*). Use this process only if the CRS applied to the layer is a wrong one or if none was applied.
- *Provider Feature Filter*: allows to restrict data to use in the current project to a subset, using functions and expression at the PDAL data provider level. Press the *Query Builder* button at the bottom to start setting the filter.



16.3 Fig. : Filtering point cloud layer to buildings at specific altitude

In the bottom part of the dialog, you can type a *Provider specific filter expression*. Such an expression can be built thanks to:

- *Fields*: the list contains all the attributes of the point cloud layer. To add an attribute to the expression, double-click its name or just type it into the text box.
- *Values*: frame lists values or statistics of the currently selected attribute, as a `key:value` pair. To add a value to the expression field, double click its name in the list: depending on the type of attribute, either the key or the value will be added to the expression. Or type the value in the expression text box.
- *Operators*: this toolbar contains all usable operators. To add an operator to the expression field, click the appropriate button. Relational operators (= , > , …) and logical operators (`AND`, `OR`, …) are available.

The *Test* button helps you check the syntax of your query. Use the *Clear* button to wipe the query and revert the layer to its original state (i.e., fully load all the points in the layer). It is possible to *Save…* the query as a `.QQF` file, or *Load…* the query from a file into the dialog.

When a filter is applied, QGIS treats the resulting subset as if it were the entire layer. For example if you applied the *filter above* for filtering buildings, you can not e.g. display, query, save or edit points that are of vegetation classification because they are not part of the subset.

Patarimas

Filtered layers are indicated in the Layers Panel

In the *Layers* panel, a filtered layer is listed with a Filter icon next to it indicating the query used when the mouse hovers over the icon. Double-click the icon to open the *Query Builder* dialog for edit. This can

also be achieved through the *Layer* ► *Filter…* menu.

16.2.3 Symbology Properties

In the *Symbology* tab the settings for the rendering of the point cloud are made. In the upper part, the settings of the different feature renderers can be found. In the lower part, there are sections with which general settings for the entire layer can be made and which apply over feature renderers.

Feature Rendering types

There are different options for rendering point clouds that can be selected using the drop-down menu at the top of the *Symbology* tab (see Fig. 16.4):

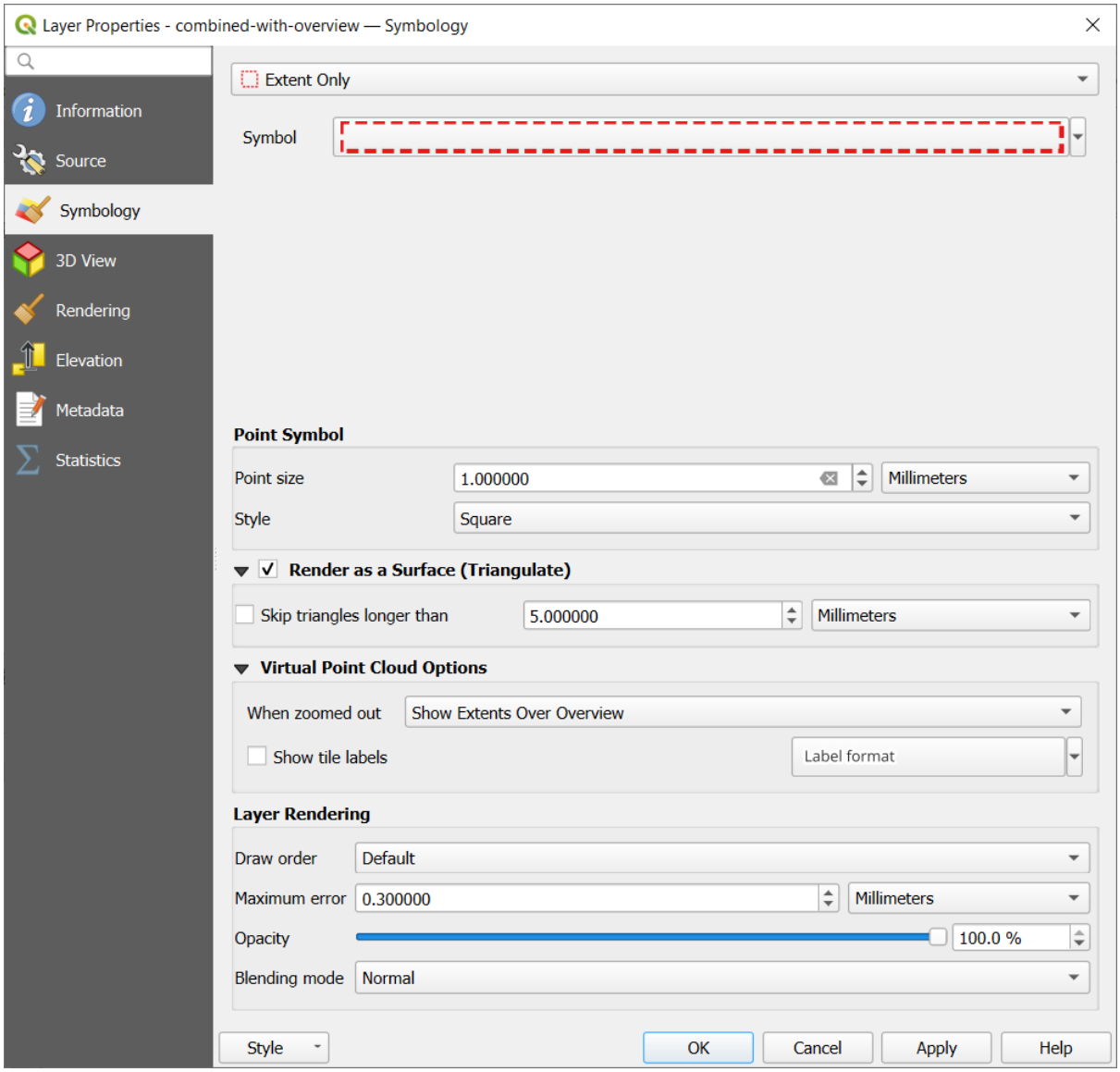
- *Extent Only*: Only a bounding box of the extent of the data is displayed; convenient for overviewing the data extent. As usual, the *Symbol widget* helps you configure any properties (color, stroke, opacity, sub-layers, …) you'd like for the box.
- *Attribute by Ramp*: The data is drawn over a color gradient. See *Attribute by Ramp Renderer*
- *RGB*: Draw the data using red, green and blue color values. See *RGB Renderer*
- *Classification*: The data is drawn using different colors for different classes. See *Classification Renderer*

When a point cloud is loaded, QGIS follows a logic to select the best renderer:

- if the dataset contains color information (red, green, blue attributes), the RGB renderer will be used
- else if the dataset contains a `Classification` attribute, the classified renderer will be used
- else it will fall back to rendering based on Z attribute

If you do not know the attributes of the point cloud, the *Statistics tab* provides a good overview of which attributes are contained in the point cloud and in which ranges the values are located.

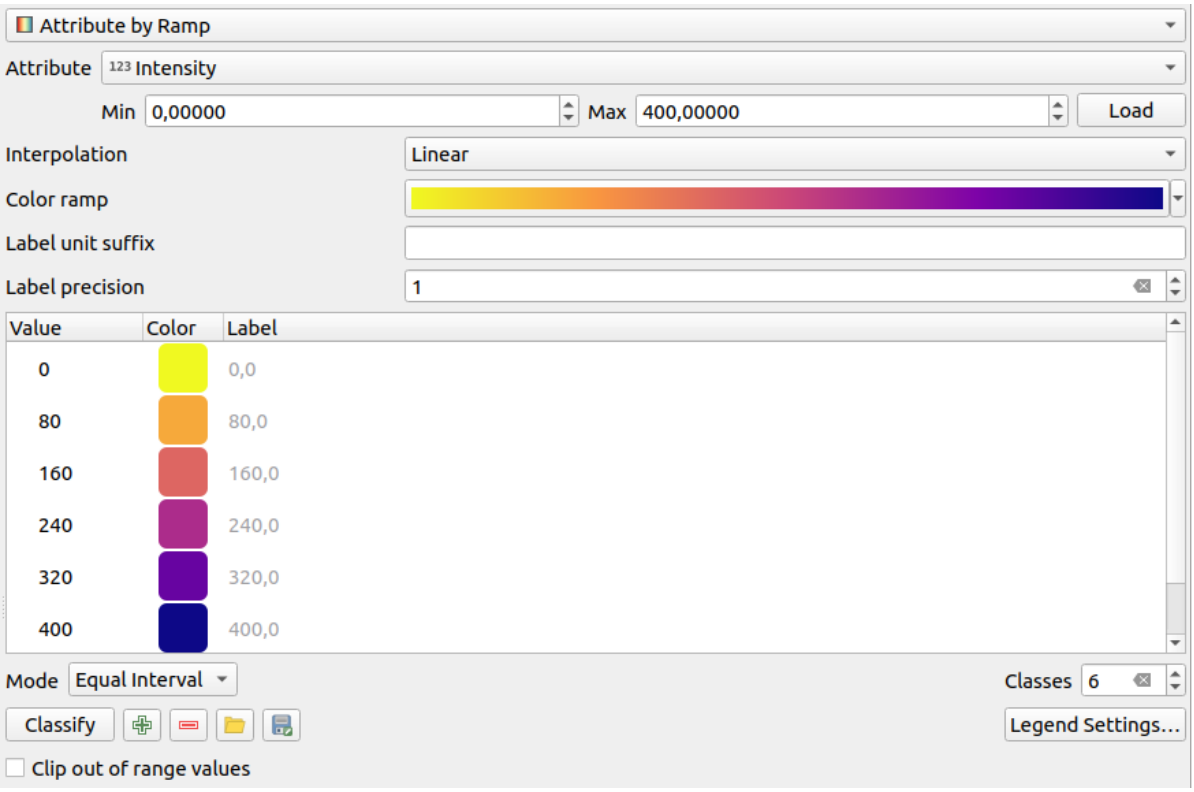
For each renderer, you can improve the data display adjusting the *point symbol size* or enabling *surface triangulation*.



16.4 Fig. : Point cloud symbology tab

Attribute by Ramp Renderer

With *Attribute by Ramp*, the data can be displayed by numerical values over a color gradient. Such numerical values can be, for example, an existing intensity attribute or the Z-value. Depending on a minimum and a maximum value, the other values are spread to the color gradient via interpolation. The distinct values and their assignment to a certain color are called „color map“ and are shown in the table. There are various setting options, which are described below the figure.



16.5 Fig. : Point cloud symbology tab: Attribute by Ramp

- *Min* and *Max* define the range that is applied to the color ramp: the *Min* value represents the left, the *Max* value the right end of the color ramp, the values in between are interpolated. By default QGIS detects the minimum and the maximum from the selected attribute but they can be modified. Once you have changed the values, you can restore the defaults by clicking on the *Load* button.
- The *Interpolation* entry defines how values are assigned their color:
  - *Discrete* (a <= symbol appears in the header of the *Value* column): The color is taken from the closest color map entry with equal or higher value
  - *Linear* The color is linearly interpolated from the color map entries above and below the pixel value, meaning that to each dataset value corresponds a unique color
  - *Exact* (a = symbol appears in the header of the *Value* column): Only pixels with value equal to a color map entry are applied a color; others are not rendered.
- The *Color ramp* widget helps you select the color ramp to assign to the dataset. As usual with *this widget*, you can create a new one and edit or save the currently selected one.
- The *Label unit suffix* adds a label after the value in the legend, and the *Label precision* controls the number of decimals to display.

The classification *Mode* helps you define how values are distributed across the classes:

- *Continuous*: Classes number and color are fetched from the color ramp stops; limits values are set following stops distribution in the color ramp (you can find more information on stops in *Spalvų rampos kūrimas*).
- *Equal interval*: The number of classes is set by the *Classes* field at the end of the line; limits values are defined so that the classes all have the same magnitude.

The classes are determined automatically and shown in the color map table. But you can also edit these classes manually:

- Double clicking in a *Value* in the table lets you modify the class value

- Double clicking in the *Color* column opens the *Spalvų parinkiklis* widget, where you can select a color to apply for that value
- Double clicking in the *Label* column to modify the label of the class
- Right-clicking over selected rows in the color table shows a contextual menu to *Change Color…* and *Change Opacity…* for the selection

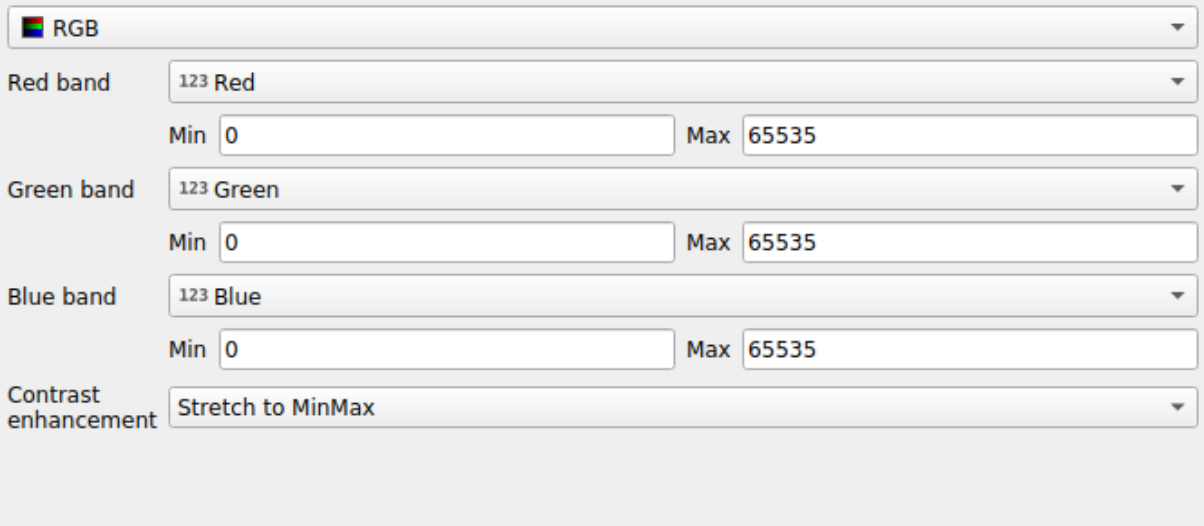
Below the table there are the options to restore the default classes with *Classify* or to manually Add values or Delete selected values from the table.

Since a customized color map can be very complex, there is also the option to Load an existing color map or to Save it for use in other layers (as a `txt` file).

If you have selected *Linear* for *Interpolation*, you can also configure:

- *Clip out of range values* By default, the linear method assigns the first class (respectively the last class) color to values in the dataset that are lower than the set *Min* (respectively greater than the set *Max*) value. Check this setting if you do not want to render those values.
- *Legend settings*, for display in the *Layers* panel and in the *layout legend*. Customization works the same way as with a raster layer (find more details at *Customize raster legend*).

RGB Renderer



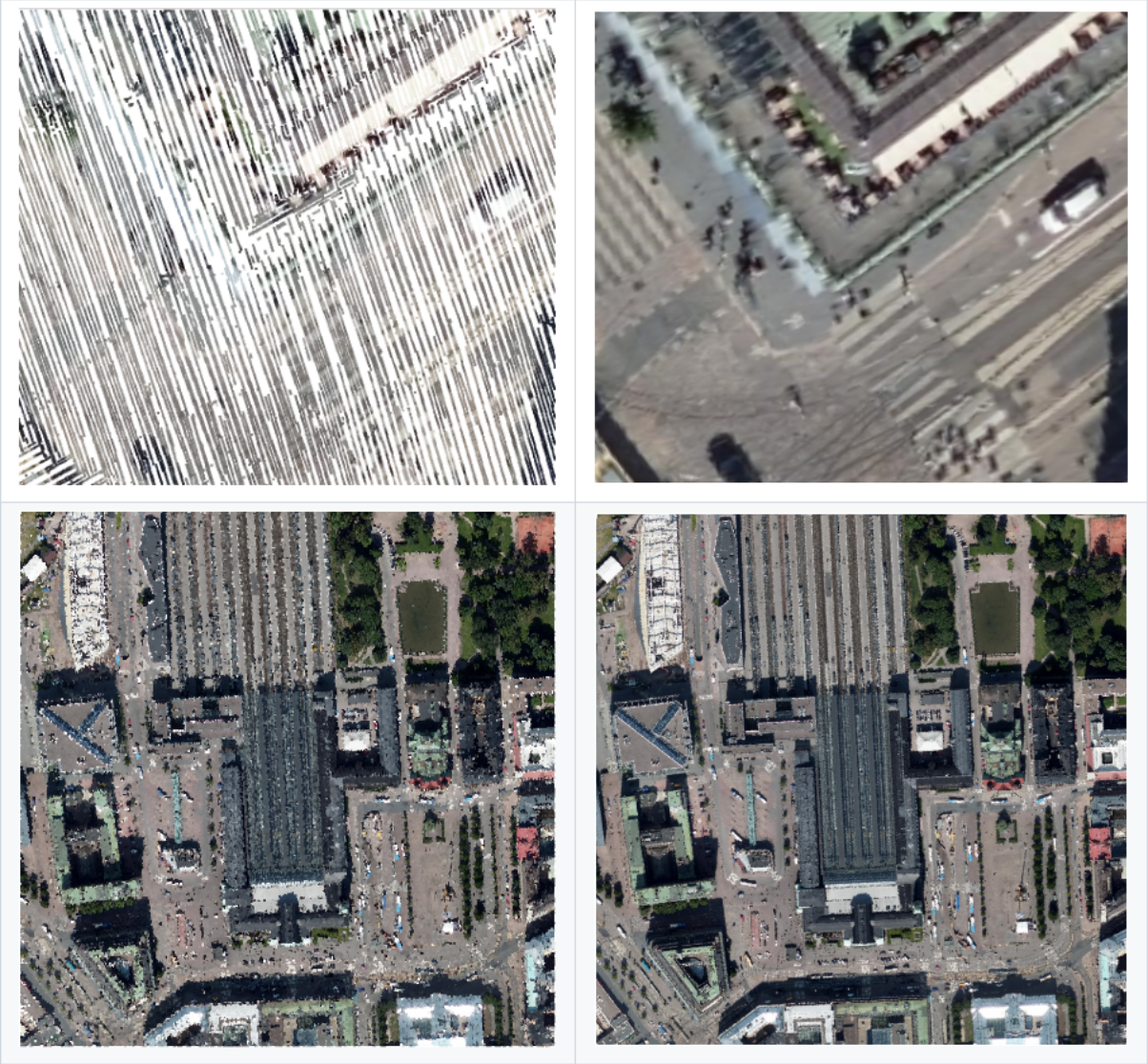
16.6 Fig. : The point cloud RGB renderer

With the *RGB* renderer, three selected attributes from the point cloud will be used as the red, green and blue component. If the attributes are named accordingly, QGIS selects them automatically and fetches *Min* and *Max* values for each band and scales the coloring accordingly. However, it is also possible to modify the values manually.

A *Contrast enhancement* method can be applied to the values: *No Enhancement*, *Stretch to MinMax*, *Stretch and Clip to MinMax* and *Clip to MinMax*

Pastaba

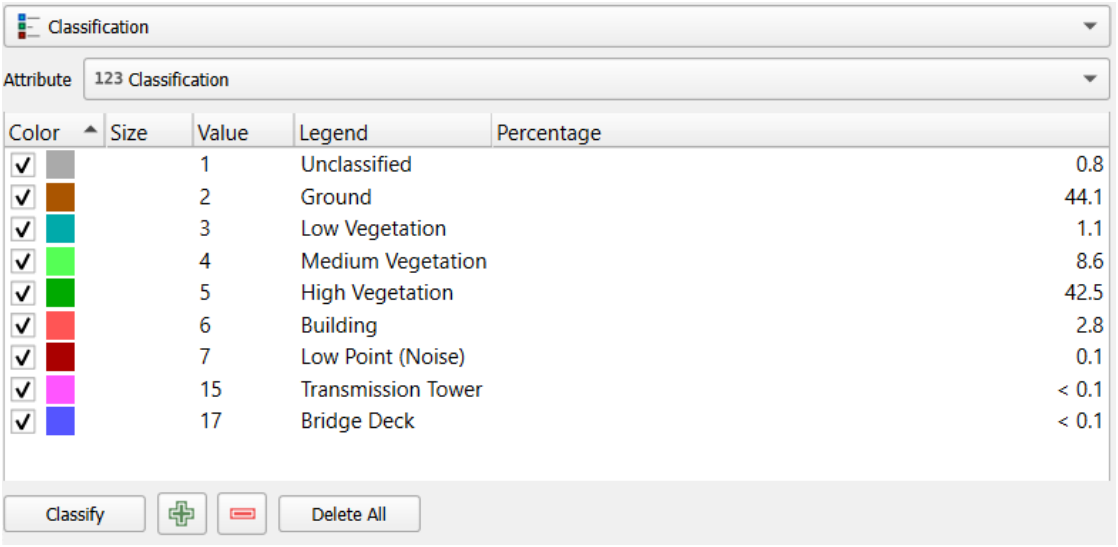
The *Contrast enhancement* tool is still under development. If you have problems with it, you should use the default setting *Stretch to MinMax*.



16.7 Fig. : Example of RGB renderer (left) combined with surface triangulation option (right)

Classification Renderer

In the *Classification* rendering, the point cloud is shown differentiated by color on the basis of an attribute. Any type of attribute can be used (numeric, string, …). Point cloud data often includes a field called `Classification`. This usually contains data determined automatically by post-processing, e.g. about vegetation. With *Attribute* you can select the field from the attribute table that will be used for the classification. By default, QGIS uses the definitions of the LAS specification (see table ‚ASPRS Standard Point Classes' in the PDF on ASPRS home page). However, the data may deviate from this schema; in case of doubt, you have to ask the person or institution from which you received the data for the definitions.



16.8 Fig. : The point cloud classification renderer

In the table all used values are displayed with the corresponding color and legend. At the beginning of each row there is a check box; if it is unchecked, this value is no longer shown on the map. With double click in the table, you can modify:

- the *Color*: the *Spalvų parinkiklis* widget opens
- the *Size*: assigning a size of `0` to a category will revert it to use the default *point size* set for the layer
- the *Value*
- the *Legend*
- the *Percentage* column will show you the representation of the category within the layer.

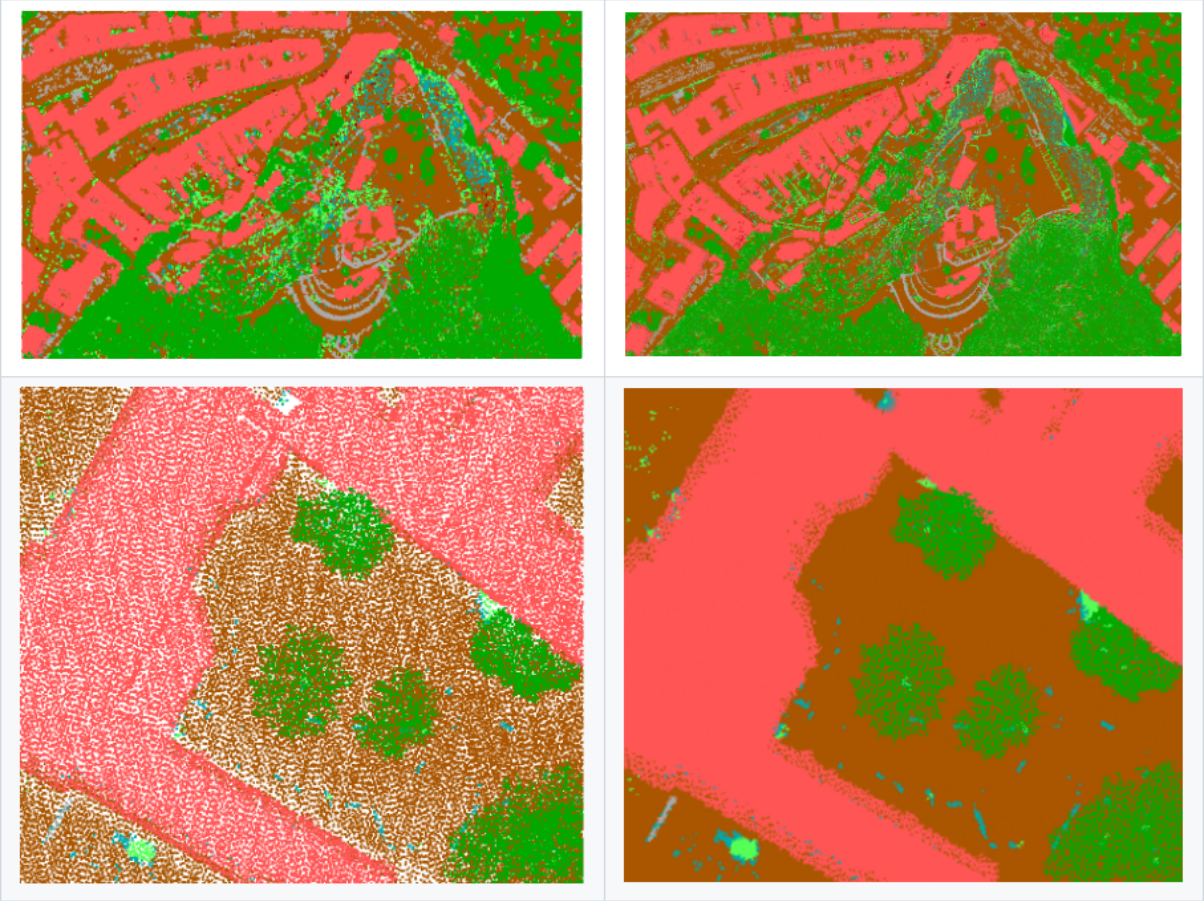
You can also right-click one or more rows to open a context menu with the options to change the color, opacity or point size.

Below the table there are buttons with which you can change the default classes generated by QGIS:

- With the *Classify* button the data can be classified automatically: all values that occur in the attributes and are not yet present in the table are added
- With Add and Delete, values can be added or removed manually
- *Delete All* removes all values from the table

Patarimas

In the *Layers* panel, you can right-click over a class leaf entry of a layer to quickly configure visibility of the corresponding features.



16.9 Fig. : Example of classification renderer (left) combined with surface triangulation option (right)

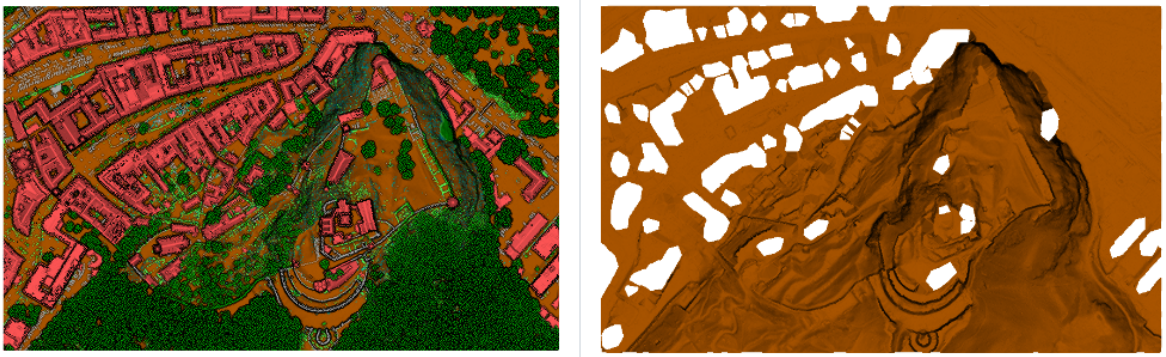
Point Symbol

Under *Point Symbol*, the size and the unit (e.g. millimeters, pixels, inches) with which each data point is displayed can be set. Either *Circle* or *Square* can be selected as the style for the points.

Render as a surface (Triangulate)

Check *Render as surface (Triangulate)* to enable the triangulation of the point cloud layer in the 2D view. This option allows rendering triangles instead of points. Each point keeps its color for interpolation in the triangle. You can control the horizontal length of computed triangles:

By checking the *Skip triangles longer than* option and setting up the threshold value, you can control the maximum lenght of a side of the triangles to consider in the horizontal plan. This can be particularly useful if you want to identify actual holes in the data.



16.10 Fig. : Rendering data as a surface with map shading (left) and with map shading, filtering large triangles (right)

Virtual Point Cloud Options

The *Virtual Point Cloud Options* are available only when the layer is a *virtual point cloud (VPC)*. QGIS renders the overview of the VPC when zoomed out, if an overview is present. This provides a seamless transition from the overview to the detailed display of individual point clouds as you zoom in.

You can control how the VPC is displayed when zoomed out using the available options:

- *Show Extents Only*: Only the extents of the underlying point clouds are displayed.
- *Show Overview Only*: Only the overview is displayed (if available).
- *Show Extents Over Overview*: The extents are displayed on top of the overview.

You can also choose to *Show tile labels* to display the tile names and set the label format.

Layer Rendering

In the *Layer Rendering* section you have the following options to modify the rendering of the layer:

- *Draw order*: allows to control whether point clouds rendering order on 2d map canvas should rely on their Z value. It is possible to render :
  - with the *Default* order in which the points are stored in the layer,
  - from *Bottom to top* (points with larger Z values cover lower points giving the looks of a true ortho photo),
  - or from *Top to bottom* where the scene appears as viewed from below.
- *Maximum error*: Point clouds usually contains more points than are needed for the display. By this option you set how dense or sparse the display of the point cloud will be (this can also be understood as ‚maximum allowed gap between points'). If you set a large number (e.g. 5 mm), there will be visible gaps between points. Low value (e.g. 0.1 mm) could force rendering of unnecessary amount of points, making rendering slower (different units can be selected).
- *Opacity*: You can make the underlying layer in the map canvas visible with this tool. Use the slider to adapt the visibility of your layer to your needs. You can also make a precise definition of the percentage of visibility in the menu beside the slider.
- *Blending mode*: You can achieve special rendering effects with this tool. The pixels of your overlaying and underlying layers are mixed through the settings described in *Suliejimo režimai*.
- *Eye dome lighting*: this applies shading effects to the map canvas for a better depth rendering. Rendering quality depends on the *draw order* property; the *Default* draw order may give sub-optimal results. Following parameters can be controlled:
  - *Stiprumas*: didina kontrastą, taip leidžiant geresnį gylio suvokimą
  - *Atstumas*: reprezentuoja naudotų taškų atstumą nuo taško centro ir padaro kraštus storesniais.

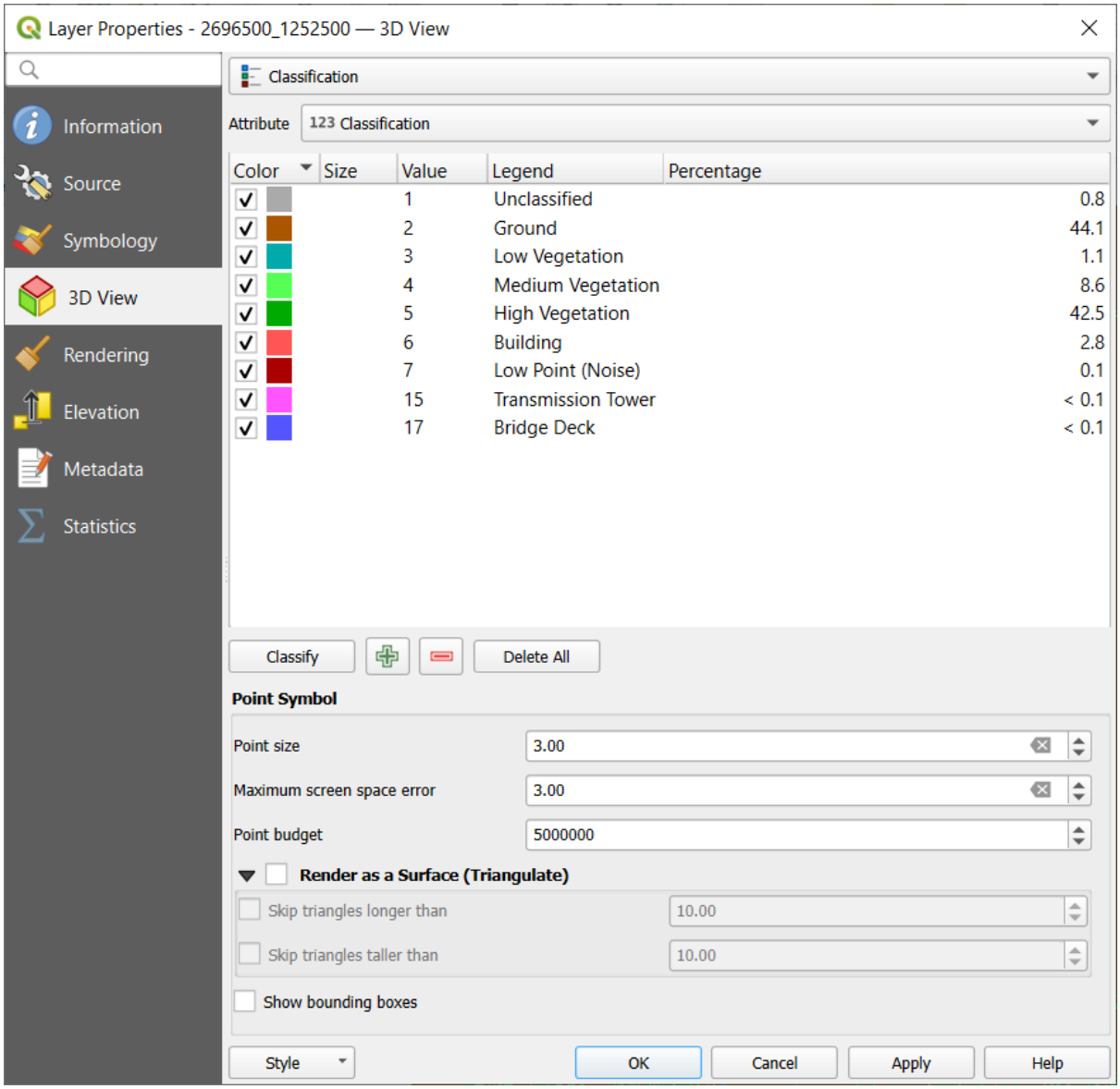
16.2.4 3D View Properties

In the *3D View* tab you can make the settings for the rendering of the point cloud in 3D maps.

3D Rendering modes

Following options can be selected from the drop down menu at the top of the tab:

- *No Rendering*: Data are not displayed
- *Follow 2D Symbology*: Syncs features rendering in 3D with *symbology assigned in 2D*
- *Single Color*: All points are displayed in the same *color* regardless of attributes
- *Attribute by Ramp*: Interpolates a given attribute over a color ramp and assigns to features their matching color. See *Attribute by Ramp Renderer*.
- *RGB*: Use different attributes of the features to set the Red, Green and Blue color components to assign to them. See *RGB Renderer*.
- *Classification*: differentiates points by color on the basis of an attribute. See *Classification Renderer*.



16.11 Fig. : The point cloud 3D view tab with the classification renderer

3D Point Symbol

In the lower part of the *3D View* tab you can find the *Point Symbol* section. Here you can make general settings for the entire layer which are the same for all renderers. There are the following options:

- *Point size*: The size (in pixels) with which each data point is displayed can be set
- *Maximum screen space error*: By this option you set how dense or sparse the display of the point cloud will be (in pixels). If you set a large number (e.g. 10), there will be visible gaps between points; low value (e.g. 0) could force rendering of unnecessary amount of points, making rendering slower (you can find more details at *Symbology* *Maximum error*).
- *Point budget*: To avoid long rendering, you can set the maximum number of points that will be rendered
- Check *Render as surface (Triangulate)* to render the point cloud layer in the 3D view with a solid surface obtained by triangulation. You can control dimensions of the computed triangles:
  - *Skip triangles longer than* a threshold value: sets in the horizontal plan, the maximum length of a side

of the triangles to consider

  - *Skip triangles taller than* a threshold value: sets in the vertical plan, the maximum height of a side of the triangles to consider

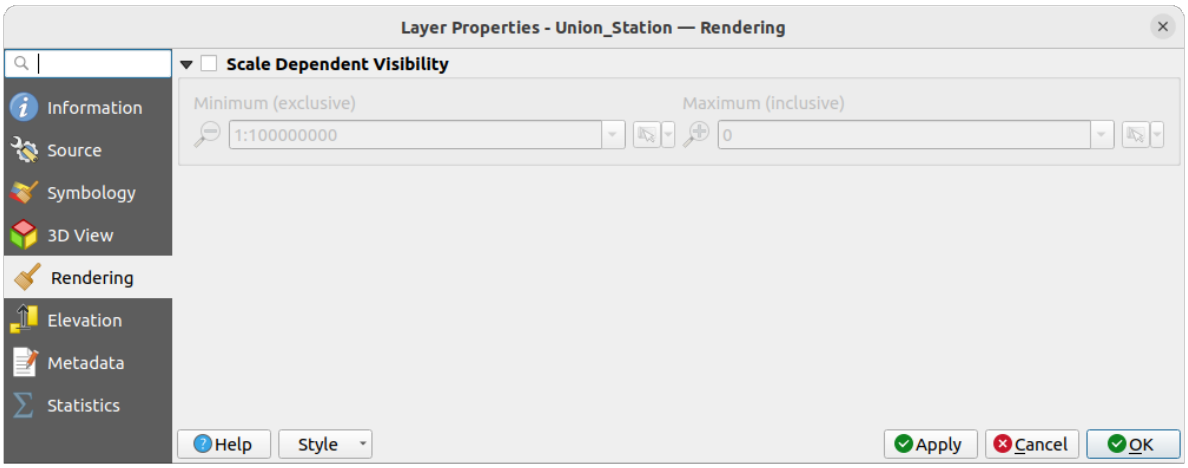
- *Show bounding boxes*: Especially useful for debugging, shows bounding boxes of nodes in hierarchy

16.2.5 Rendering Properties

Under the *Scale dependent visibility* group box, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scale, defining a range of scale in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See *Nuo mastelio priklausomo matomumo parinkimas* for more information.

Pastaba

You can also activate scale dependent visibility on a layer from within the *Layers* panel: right-click on the layer and in the contextual menu, select *Set Layer Scale Visibility*.



16.12 Fig. : The point cloud rendering tab

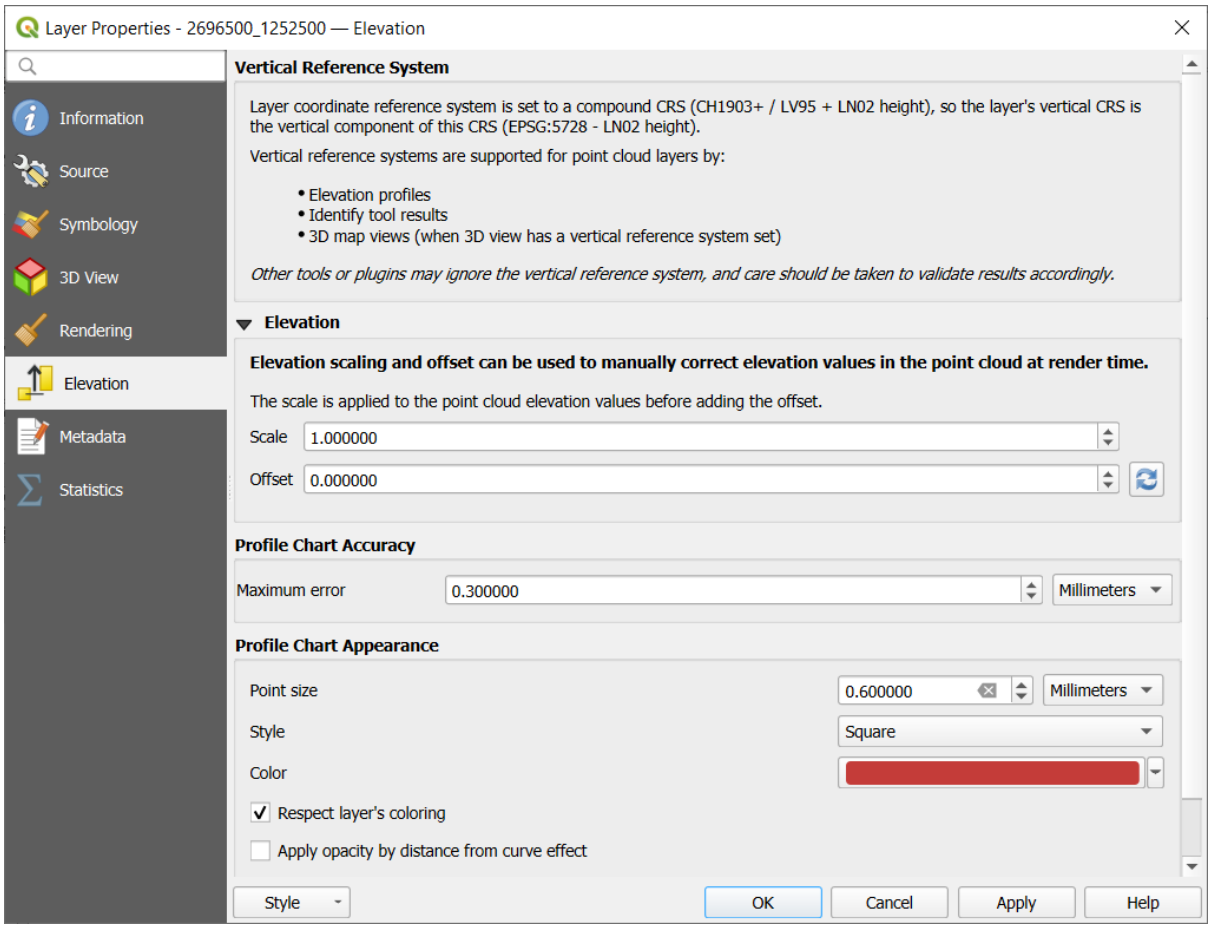
16.2.6 Aukščio savybės

In the *Elevation* tab, you can set corrections for the Z-values of the data. This may be necessary to adjust the elevation of the data in 3D maps and its appearance in the *profile tool charts*. There are following options:

- *Vertical Reference System*: If the CRS of your point cloud layer is a compound one (including a Z dimension), then the vertical CRS used for the layer will be automatically derived from the vertical component of the layer's CRS. In this case, you cannot manually set a different vertical CRS, and the option to change it will be disabled. If your point cloud layer uses a horizontal (2D) CRS (though uncommon), you can manually select a specific vertical CRS by clicking on the Select CRS. Vertical reference systems are supported for point cloud layers in:
  - *Elevation profiles*
  - *Identify Tool Results*
  - *3D map views*
- Under *Elevation* group:

  - You can set a *Scale*: If `10` is entered here, a point that has a value Z = `5` is displayed at a height of `50`.
  - An *offset* to the z-level can be entered. This is useful to match different data sources in its height to each other. By default, the lowest z-value contained in the data is used as this value. This value can also be restored with the Refresh button at the end of the line.

- Under *Profile Chart Accuracy*, the *Maximum error* helps you control how dense or sparse the points will be rendered in the elevation profile. Larger values result in a faster generation with less points included.
- Under *Profile Chart Appearance*, you can control the point display:
  - *Point size*: the size to render the points with, in supported units (millimeters, map units, pixels, …)
  - *Style*: whether to render the points as *Circle* or *Square*
  - Apply a single *Color* to all the points visible in the profile view
  - Check *Respect layer's coloring* to instead show the points with the color assigned via their *2D symbology*
  - *Apply opacity by distance from curve effect*, reducing the opacity of points which are further from the profile curve



16.13 Fig. : The point cloud elevation tab

16.2.7 Metaduomenų savybės

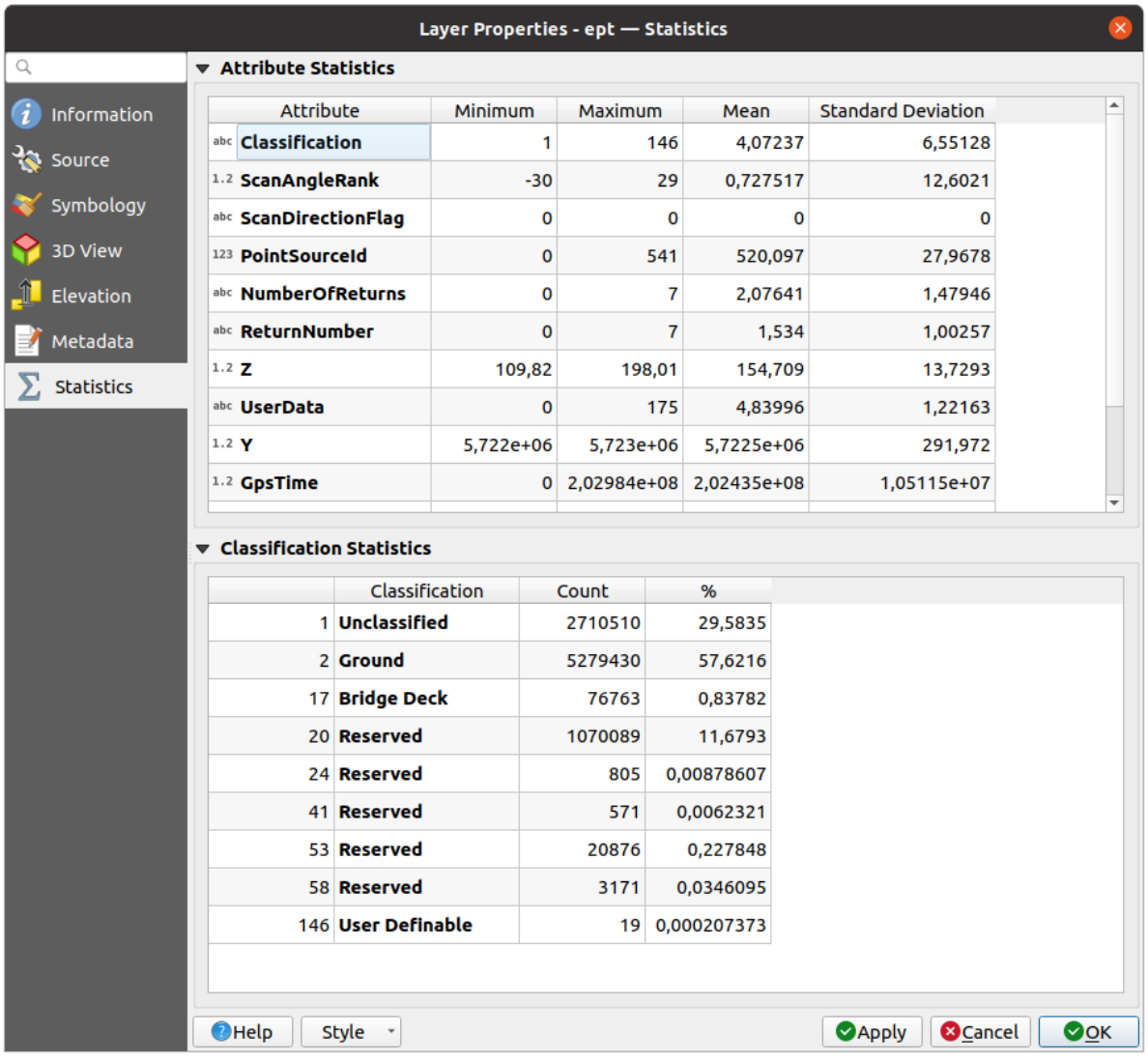
The *Metadata* tab provides you with options to create and edit a metadata report on your layer. See *Metaduomenys* for more information.

16.2.8 Statistics Properties

In the *Statistics* tab you can get an overview of the attributes of your point cloud and their distribution.

At the top you will find the section *Attribute Statistics*. Here all attributes contained in the point cloud are listed, as well as some of their statistical values: *Minimum*, *Maximum*, *Mean*, *Standard Deviation*

If there is an attribute *Classification*, then there is another table in the lower section. Here all values contained in the attribute are listed, as well as their absolute *Count* and relative *%* abundance.



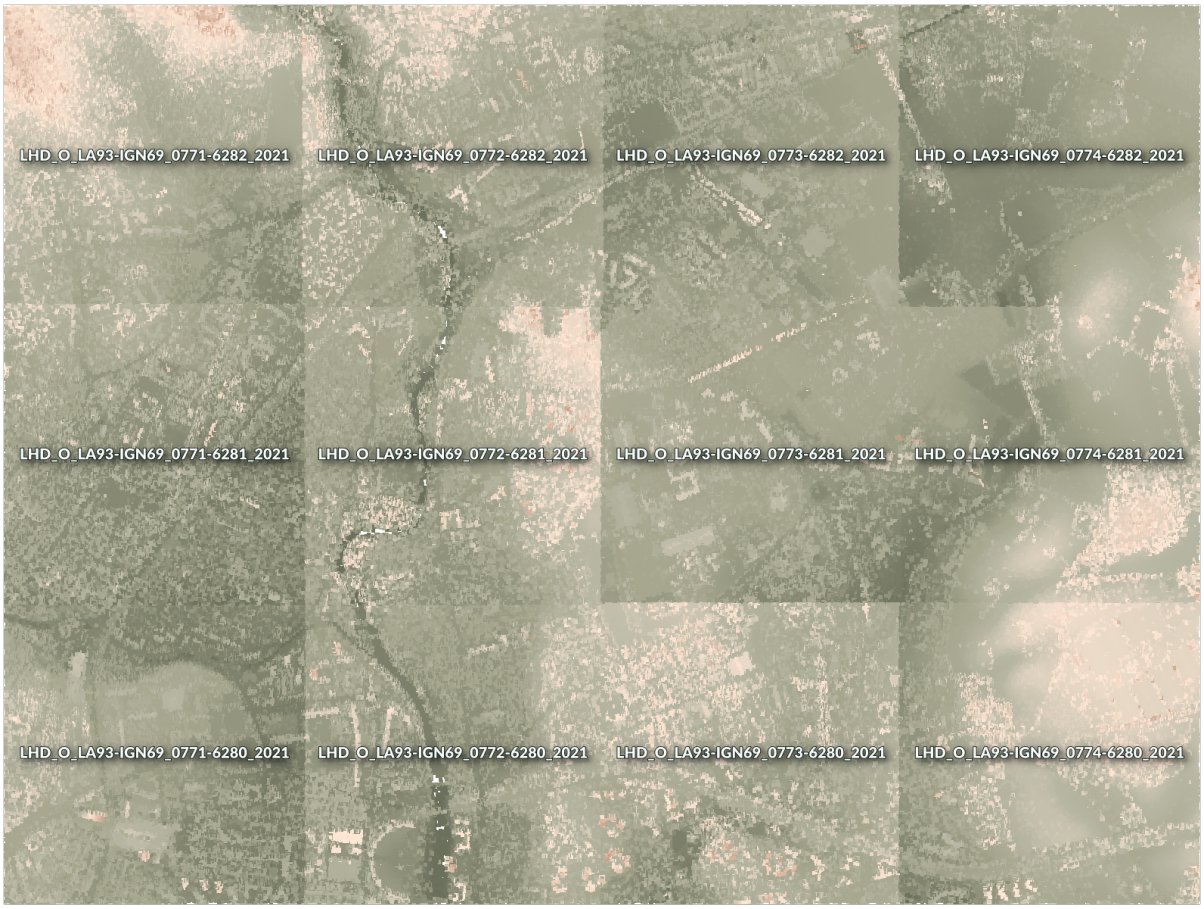
16.14 Fig. : The point cloud statistics tab

16.3 Virtual point cloud

Lidar surveys of larger areas are often multi-terabyte datasets with many billions of points. Having such large datasets represented as a single point cloud file is not practical due to the difficulties of storage, transfer, display and analysis. Point cloud data are therefore typically stored and distributed split into square tiles (e.g. `1km x 1km`), each tile having a more manageable file size (e.g. ~200 MB when compressed).

Tiling of data solves the problems with size of data, but it introduces issues when processing or viewing an area of interest that does not fit entirely into a single tile. Users need to develop workflows that take into account multiple tiles and special care needs to be taken to deal with data near edges of tiles to avoid unwanted artefacts in outputs. Similarly, when viewing point cloud data, it becomes cumbersome to load many individual files and apply the same symbology.

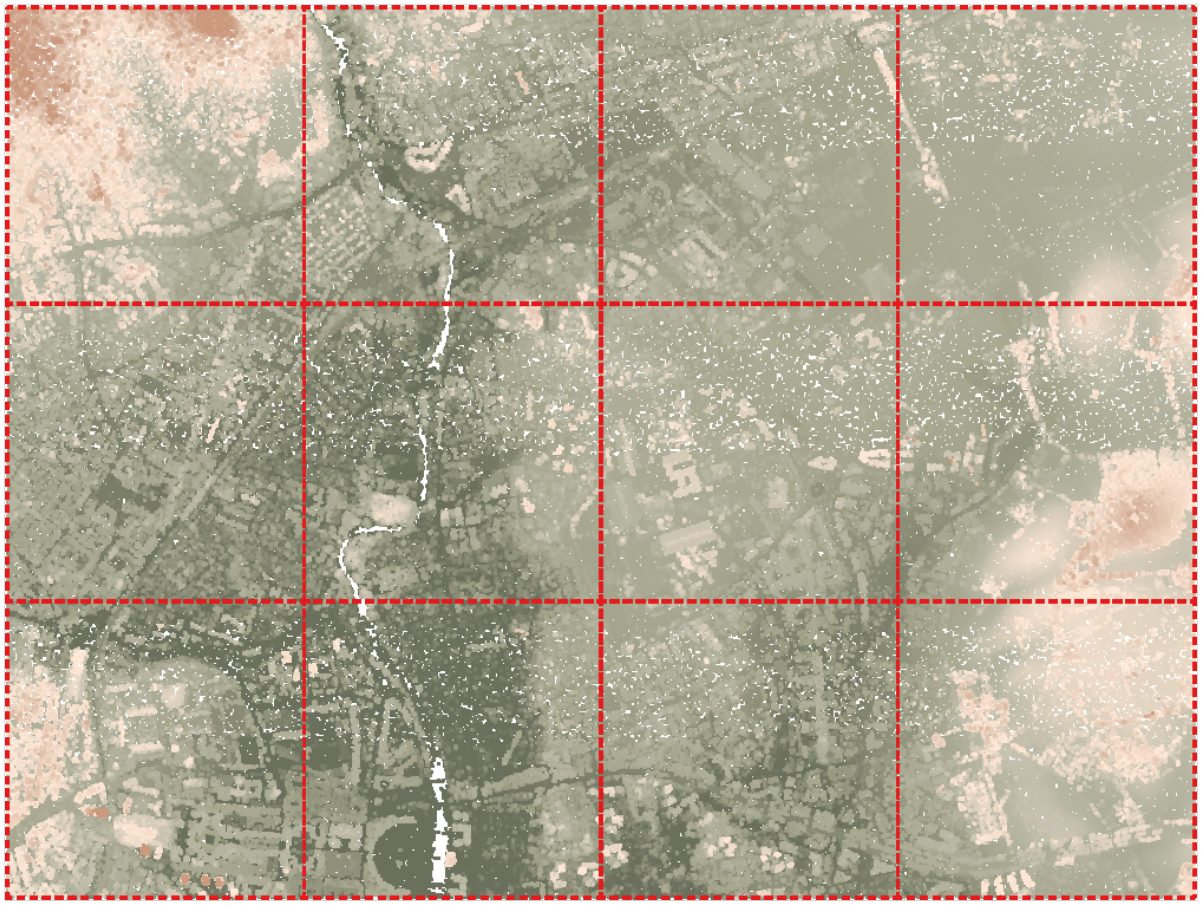
Here is an example of several point cloud tiles loaded in QGIS. Each tile is styled based on min/max Z values of the tile, creating visible artefacts on tile edges. The styling has to be adjusted for each layer separately:



16.15 Fig. : Individual point cloud tiles loaded, with artefacts on edges

In the GIS world, many users are familiar with the concept of virtual rasters. A virtual raster is a file that simply references other raster files with actual data. In this way, GIS software then treats the whole dataset comprising many files as a single raster layer, making the display and analysis of all the rasters listed in the virtual file much easier.

Borrowing the concept of virtual rasters from GDAL, **virtual point cloud (VPC)** is a file format that references other point cloud files. Software supporting virtual point clouds handles the whole tiled dataset as a single data source.



16.16 Fig. : The virtual point cloud

Displaying and manipulating virtual point cloud is much more fluent and easy.

At the core, a virtual point cloud file is a simple JSON file with `.vpc` extension, containing references to actual data files (e.g. `.LAS`, `.LAZ` or `.COPC` files) and additional metadata extracted from the files. Even though it is possible to write VPC files by hand, it is strongly recommended to create them using an automated tool, such as:

- The Processing *Kurti virtualų taškų masyvą (VPC)* algorithm
- The `build_vpc` command of PDAL wrench tool

For more details, please refer to the VPC specification that also contains best practices and optional extensions (such as overviews).

SKYRIUS 17

Working with 3D Tiles

17.1 What are 3D Tiles?

3D tiles are specification for streaming and rendering large-scale 3D geospatial datasets. They use a hierarchical structure to efficiently manage and display 3D content, optimising performance by dynamically loading appropriate levels of detail. This technology is widely used in urban planning, architecture, simulation, gaming, and virtual reality, providing a standardised and interoperable solution for visualising complex geographical data.

17.1 Fig. : Example of 3D tiles

See *3D scenų kaladėlių paslaugų naudojimas* for instructions on how to add these data sources to QGIS.

17.1.1 State of 3D Tiles Support

Currently, QGIS supports two formats of 3D tiles:

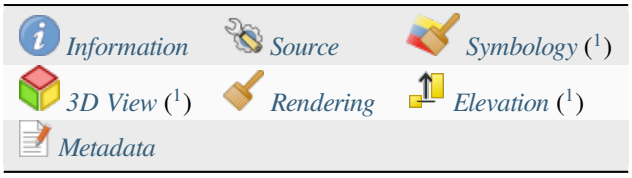
- „Cesium“ 3D Tiles are used primarily for complex 3D models of buildings or whole cities. Such datasets can be provided by cloud-based platforms such as Cesium Ion or Google (*Photorealistic 3D Tiles*).
- Quantized Mesh tiles are used for terrain elevation data.

Supported *3D Tiles* features:

- QGIS currently has partial support for 3D Tiles 1.0 and partial support for 3D Tiles 1.1.
- For 3D Tiles 1.0, the tile format Batched 3D Model (`b3dm`) is supported. The tile formats Instanced 3D Model (`i3dm`), Point Cloud (`pnts`) and Composite (`cmpt`) are not supported yet.
- For 3D Tiles 1.1, the `3DTILES_content_gltf` extension is supported, so tile contents may also be encoded as glTF 2.0 (text or binary). The `EXT_mesh_gpu_instancing` extension is not supported yet.
- Currently only explicit tiling is supported. Implicit tiling is not supported yet
- Currently only tilesets using EPSG:4978 are supported.

17.2 3D Tiles Properties

The 3D tiles *Layer Properties* dialog provides the following sections:



17.2.1 Informacinės savybės

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata on the current layer. Provided information are:

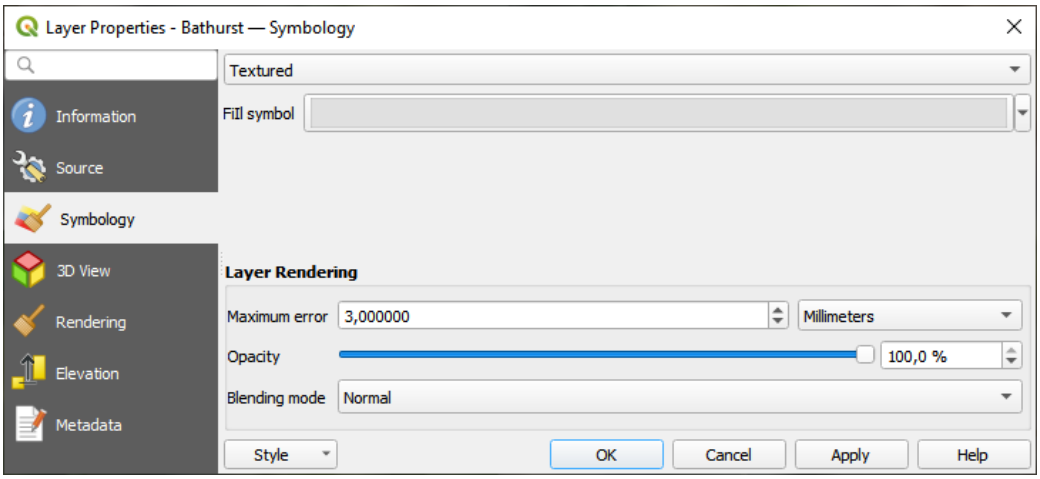
- based on the provider of the layer: name, URL, source type and path, number of zoom levels
- custom properties, used to store in the active project additional information about the layer. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- the Coordinate Reference System: name, units, method, accuracy, reference (i.e. whether it's static or dynamic)
- picked from the *filled metadata*: access, extents, links, contacts, history…

17.2.2 Source Properties

The *Source* tab displays basic information about the selected 3D tile, including:

- the *Layer name* to display in the *Layers Panel*;
- the *Coordinate Reference System*: Displays the layer's *Coordinate Reference System (CRS)*. You can change the layer's CRS, by selecting a recently used one in the drop-down list or clicking on the Select CRS button (see *Koordinačių atskaitos sistemos parinkiklis*). Use this process only if the layer CRS is wrong or not specified.

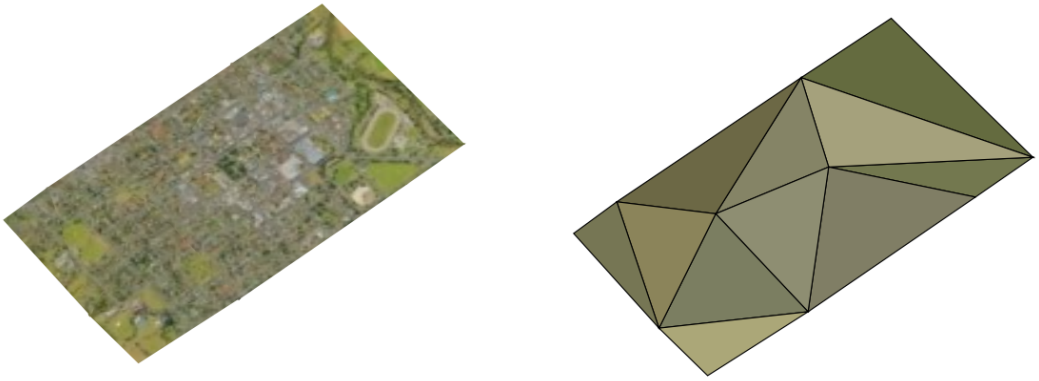
17.2.3 Symbology Properties



17.2 Fig. : 3D Tile Layer Symbology

[1] Also available in the *Layer styling panel*

By default, the layer is styled using texture, but you can change it to see the wireframe mesh behind the scene by choosing *Wireframe* in the drop-down menu. You can also, change the mesh fill and line symbols similar to the vector polygons. Checking *Use texture colors* will render each mesh element with the average value of the full texture. This is a good option to try when dealing with a large dataset and want to get a quick overview of the data.



17.3 Fig. : 3D Tiles - textured and wireframe

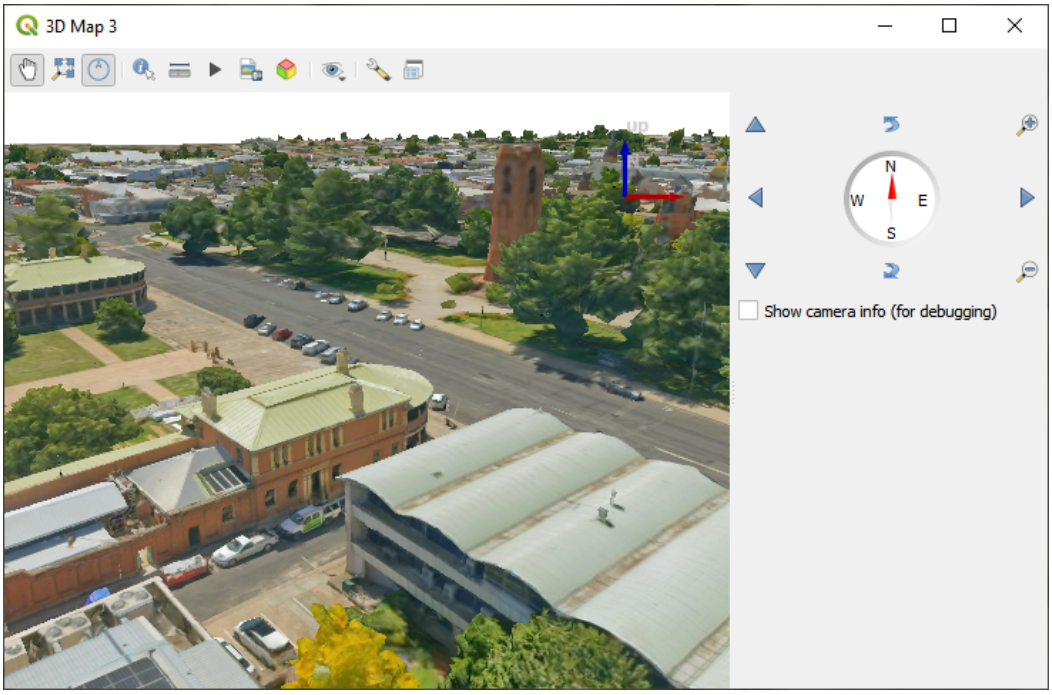
From the *Symbology* tab, you can also set some options that invariably act on all features of the layer:

- *Maximum error*: This parameter determines the level of detail displayed in the 3D model. Similar to point clouds, 3D tiles often contain more information than necessary for visual representation. By adjusting this setting, you control the balance between display density and rendering speed. A larger value (e.g., 5 mm) may introduce noticeable gaps between elements, while a smaller value (e.g., 0.1 mm) could lead to the rendering of an excessive number of details, potentially slowing down the rendering process. Different units can be selected to tailor the setting to your specific needs.
- *Opacity*: Adjusts the visibility of the underlying layer on the map canvas using this tool. Use slider to tailor the visibility of your scene layer according to your preferences. Alternatively, specify the exact percentage of visibility through the menu next to the slider.
- *Blending mode*: You can achieve special rendering effects with these tools that you may previously only know from graphics programs. The pixels of your overlaying and underlaying layers are mixed through the settings described in *Suliejimo režimai*.

17.2.4 3D View Properties

- *Maximum screen space error*: Determines the threshold for swapping terrain tiles with more detailed ones (and vice versa) - i.e. how soon the 3D view will use higher quality tiles. Lower numbers mean more details in the scene at the expenses of increased rendering complexity.
- *Show bounding boxes*: Shows 3D bounding boxes of the terrain tiles (useful for troubleshooting terrain issues).

To view the data you can open *New 3D map view*.



17.4 Fig. : 3D žemėlapio vaizdas

17.2.5 Rendering Properties

Under *Scale dependent visibility*, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scales, defining a range of scales in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See *Nuo mastelio priklausomo matomumo parinkimas* for more information.

17.2.6 Aukščio savybės

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view*. Specifically, you can set:

- *Elevation Surface*: how the 3D layer vertices Z values should be interpreted as terrain elevation. You can apply a *Scale* factor and an *Offset*.

17.2.7 Metaduomenų savybės

The *Metadata* tab provides you with options to create and edit a metadata report on your layer. See *Metaduomenys* for more information.

SKYRIUS 18

Laying out the maps

With Print Layouts and Reports you can create maps and atlases, and print them or save them as image, PDF or SVG files.

18.1 Spausdinimo išdėstymo apžvalga

The print layout provides growing layout and printing capabilities. It allows you to add elements such as the QGIS 2D or 3D map canvas, text labels, images, legends, scale bars, basic shapes, arrows, attribute and simple tables, elevation profiles and HTML frames… You can size, group, align, position and rotate each element and adjust their properties to create your layout.

The layout can be printed or exported to image formats, PostScript, PDF or to SVG. You can save the layout as a template and load it again in another session. Finally, generating several maps based on a template can be done through the atlas generator.

18.1.1 Paprastos sesijos pradedantiesiems

Before you start to work with the print layout, you need to load some raster or vector layers in the QGIS map canvas and adapt their properties to suit your own convenience. After everything is rendered and symbolized to your liking, click the New Print Layout icon in the *Project* toolbar or choose *Project* ► *New Print Layout*. You will be prompted to choose a title for the new layout.

To demonstrate how to create a map please follow the next instructions.

1. On the left side, select the Add map toolbar button and draw a rectangle on the canvas holding down the left mouse button. Inside the drawn rectangle the QGIS map view to the canvas.

2. Select the Add scalebar toolbar button and click with the left mouse button on the print layout canvas. A scalebar will be added to the canvas.

3. Select the Add legend toolbar button and draw a rectangle on the canvas holding down the left mouse button. Inside the drawn rectangle the legend will be drawn.

4. Select the Select/Move item icon to select the map on the canvas and move it a bit.

5. While the map item is still selected you can also change the size of the map item. Click while holding down the left mouse button, in a white little rectangle in one of the corners of the map item and drag it to a new location to change its size.
6. Click the *Item Properties* panel on the left down side and find the setting for the orientation. Change the value of the setting *Map orientation* to ‚15.00°‘. You should see the orientation of the map item change.
7. Now, you can print or export your print layout to image formats, PDF or to SVG with the export tools in *Layout* menu.
8. Finally, you can save your print layout within the project file with the Save Project button.

You can add multiple elements to the print layout. It is also possible to have more than one map view or legend or scale bar in the print layout canvas, on one or several pages. Each element has its own properties and, in the case of the map, its own extent. If you want to remove any elements from the layout canvas, you can do that with the `Delete` or the `Backspace` key.

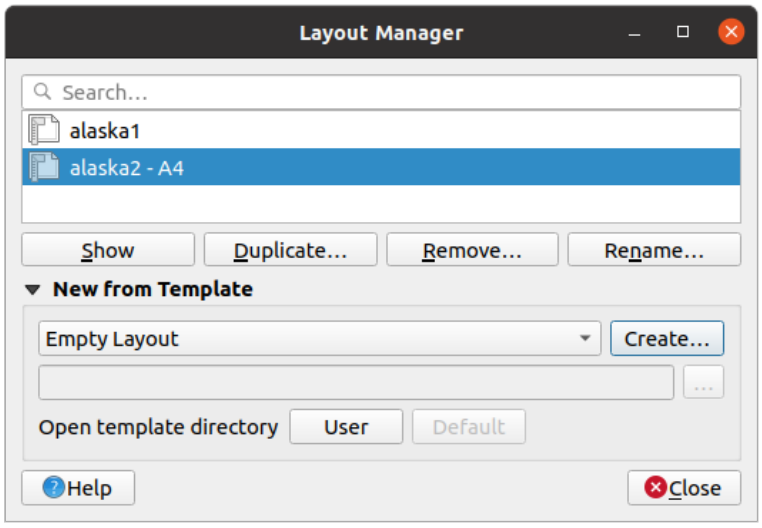
18.1.2 The Layout Manager

The *Layout Manager* is the main window to manage print layouts in the project. It gives you an overview of existing print layouts and reports in the project and offers tools to:

- search for a layout;
- add new print layout or new report from scratch, template or duplicating an existing one;
- rename or delete any of them;
- open them in the project.

To open the layout manager dialog:

- from the main QGIS dialog, select *Project* ► *Layout Manager…* menu or click on the Layout Manager button in the *Project Toolbar*;
- from a print layout or report dialog, select *Layout* ► *Layout Manager…* menu or click on the Layout Manager button in the *Layout Toolbar*.



18.1 Fig. : The Print Layout Manager

The layout manager lists in its upper part all the available print layouts or reports in the project with tools to:

- show the selection: you can select multiple reports and/or print layout(s) and open them in one-click. Double-click a name also opens it;

- duplicate the selected print layout or report (available only if one item is selected): it creates a new dialog using the selected one as template. You'll be prompted to choose a new title for the new layout;
- rename the report or layout (available only if one item is selected): you'll be prompted to choose a new title for the layout;
- remove the layout: the selected print layout(s) will be deleted from the project.

In the lower part, it's possible to create new print layouts or reports from scratch or a template. By default, QGIS will look for templates in the *user profile* folder and the application template directories (accessible with the two buttons at the bottom of the frame) but also in any folder declared as *Path(s) to search for extra print templates* in *Settings* ► *Options* ► *Layouts*. Found templates are listed in the combobox. Select an item and press the *Create* button to generate a new report or print layout.

You can also use layout templates from a custom folder; in that case, select *specific* in the templates drop-down list, browse to the template and press *Create*.

Patarimas

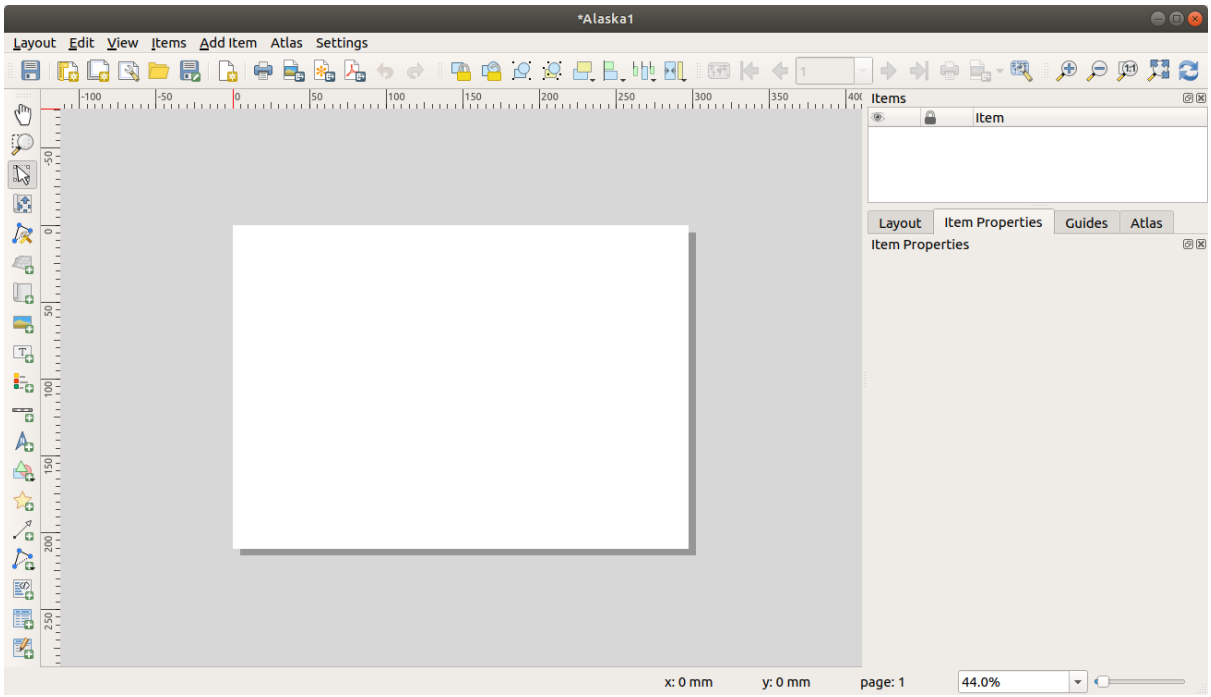
Creating template-based print layouts from Browser panel

Drag-and-drop a print layout template `.qpt` file from any file browser onto the map canvas or double-click it in the *Browser panel* generates a new print layout from the template.

18.1.3 Menus, tools and panels of the print layout

Opening the print layout provides you with a blank canvas that represents the paper surface when using the print option. Initially you find buttons on the left beside the canvas to add print layout items: 2D or 3D map canvases, text labels, images, legends, scale bars, basic shapes, arrows, attribute or simple tables, HTML frames, elevation profiles,… In this toolbar you also find buttons to navigate, zoom in on an area and pan the view on the layout as well as buttons to select any layout item and to move the extents of the map items.

Fig. 18.2 shows the initial view of the print layout before any elements are added.



18.2 Fig. : Spausdinimo išdėstymas

On the right beside the canvas you find two set of panels. The upper one holds the panels *Items* and *Undo History* and the lower holds the panels *Layout*, *Item properties* and *Atlas generation*.

- The *Items* panel provides a list of all the print layout items added to the canvas and ways to globally interact with them (see *The Items Panel* for more information).
- The *Undo History* panel displays the history of all changes applied to the layout. With a mouse click, it is possible to undo and redo layout steps back and forth to a certain status.
- The *Layout* panel allows you to set general parameters to apply to the layout when exporting or working within (see *The Layout Panel* for more details);
- The *Item Properties* panel displays the properties for the selected item. Click the Select/Move item icon to select an item (e.g., legend, scale bar or label) on the canvas. Then click the *Item Properties* panel and customize the settings for the selected item (see *Layout Items* for detailed information on each item settings).
- The *Atlas* panel allows you to enable the generation of an atlas for the current layout and gives access to its parameters (see *Generate an Atlas* for detailed information on atlas generation usage).

In the bottom part of the print layout window, you can find a status bar with mouse position, current page number, a combo box to set the zoom level, the number of selected items if applicable and, in the case of atlas generation, the number of features.

In the upper part of the print layout window, you can find menus and other toolbars. All print layout tools are available in menus and as icons in a toolbar.

The toolbars and the panels can be switched off and on using the right mouse button over any toolbar, or through *View* ► *Toolbars* ► or *View* ► *Panels* ►.

Menus and Tools

Layout menu

The *Layout* provides action to manage the layout:

- Save the project file directly from the print layout window.
- Create a new and blank print layout with *New Layout…*.
- *Duplicate Layout…* : Create a new print layout by duplicating the current one.
- Remove the current layout with *Delete Layout…*.
- Open the *Layout Manager…*.
- *Layouts* ► : Open an existing print layout.

Once the layout is designed, with *Save as Template* and *Add Items from Template* icons, you can save the current state of a print layout session as a `.qpt` template file and load its items again in another session/print layout.

In the *Layout* menu, there are also powerful ways to share geographical information produced with QGIS that can be included in reports or published. These tools are *Export as Image…*, *Export as PDF…*, *Export as SVG…* and *Print…*.

Below is a list of all the available tools in this menu with some convenient information.

18.1 Table : „Layout“ Menu Tools

Įrankis	Kombinacija	Įrankinė	Aprašymas
Save Project	`Ctrl+S`	*Layout*	*Įvadas į QGIS projektus*
New Layout	`Ctrl+N`	*Layout*	*The Layout Manager*
Duplicate Layout		*Layout*	*The Layout Manager*
Delete Layout			
Maketų tvarkyklė…		*Layout*	*The Layout Manager*
Maketai ►			
Layout Properties…			*The Layout Panel*
Rename Layout…			
Page Properties…			*Working with the page properties*
Add Pages…		*Layout*	*Working with the page properties*
Add Items from Template		*Layout*	*Creating a layout item*
Save as Template…		*Layout*	*The Layout Manager*
Export as Image…		*Layout*	*Export as Image*
Export as SVG…		*Layout*	*Export as SVG*
Export as PDF…		*Layout*	*Export as PDF*
Printer Page Setup…	`Ctrl+Shift+P`		
Print…	`Ctrl+P`	*Layout*	*Išvesties sukūrimas*
Uždaryti	`Ctrl+Q`		

Keitimo meniu

The *Edit* menu offers tools to manipulate print layout items. It includes common actions like selection tools, Copy/Cut/Paste and undo/redo (see *The Undo History Panel: Revert and Restore actions*) functionality for the items in the layout.

When using the Paste action, the elements will be pasted according to the current mouse position. Using the *Edit* ► *Paste in Place* action or pressing `Ctrl+Shift+V` will paste the items into the current page, at the same position they were in their initial page. It ensures to copy/paste items at the same place, from page to page.

Below is a list of all the available tools in this menu with some convenient information.

18.2 Table : „Edit“ Menu Tools

Įrankis	Kombinacija	Įrankinė	Aprašymas
Undo (last change)	`Ctrl+Z`	*Layout*	*The Undo History Panel: Revert and Restore actions*
Redo (last reverted change)	`Ctrl+Y`	*Layout*	*The Undo History Panel: Revert and Restore actions*
Delete	`Del`		
Iškirpti	`Ctrl+X`		
Kopijuoti	`Ctrl+C`		
Įkelti	`Ctrl+V`		
Paste in place	`Ctrl+Shift+V`		

continues on next page

Table 18.2 – tęsinys iš praeito puslapio

Įrankis	Kombinacija	Įrankinė	Aprašymas
Pažymėti viską	`Ctrl+A`		
Deselect all	`Ctrl+Shift+A`		
Invert Selection			
Select Next Item Below	`Ctrl+Alt+[`		
Select Next Item above	`Ctrl+Alt+]`		
Pan Layout	`P`	*Toolbox*	
Zoom	`Z`	*Toolbox*	
Select/Move Item	`V`	*Toolbox*	*Interacting with layout items*
Move Content	`C`	*Toolbox*	*The Map Item*
Edit Nodes Item		*Toolbox*	*The Node-Based Shape Items*

Vaizdo meniu

The *View* menu gives access to navigation tools and helps to configure general behavior of the print layout. Beside the common zoom tools, you have means to:

- Refresh view (if you find the view in an inconsistent state);
- enable a *grid* you could snap items to when moving or creating them. Grids setting is done in *Settings* ► *Layout Options…* or in the *Layout Panel*;
- enable *guides* you could snap items to when moving or creating them. Guides are red lines that you can create by clicking in the ruler (above or at the left side of the layout) and drag and drop to the desired location;
- *Smart Guides*: uses other layout items as guides to dynamically snap to as you move or reshape an item;
- *Clear Guides* to remove all current guides;
- *Show Bounding box* around the items to better identify your selection;
- *Show Rules* around the layout;
- *Show Pages* or set up pages to transparent. Often layout is used to create non-print layouts, e.g. for inclusion in presentations or other documents, and it's desirable to export the composition using a totally transparent background. It's sometimes referred to as „infinite canvas“ in other editing packages.

In the print layout, you can change the zoom level using the mouse wheel or the slider and combo box in the status bar. If you need to switch to pan mode while working in the layout area, you can hold the `Spacebar` or the mouse wheel. With `Ctrl+Spacebar`, you can temporarily switch to Zoom In mode, and with `Ctrl+Alt+Spacebar`, to Zoom Out mode.

Panels and toolbars can be enabled from the *View* ► menu. To maximise the space available to interact with a composition you can check the *View* ► *Toggle Panel Visibility* option or press `Ctrl+Tab`; all panels are hidden and only previously visible panels are restored when unchecked.

It's also possible to switch to a full screen mode to have more space to interact with by pressing `F11` or using *View* ► *Toggle Full Screen*.

18.3 Table : „View“ Menu Tools

Įrankis	Kombinacija	Įrankinė	Aprašymas
Atnaujinti	`F5`	*Navigation*	

continues on next page

Table 18.3 – tęsinys iš praeito puslapio

Įrankis	Kombinacija	Įrankinė	Aprašymas
Preview ►			
Didinti	`Ctrl++`	*Navigation*	
Mažinti	`Ctrl+-`	*Navigation*	
Didinti iki 100%	`Ctrl+1`	*Navigation*	
Pilnai padidinti	`Ctrl+0`	*Navigation*	
Zoom to Width			
Show Grid	`Ctrl+'`		*Guides and Grid*
Snap to Grid	`Ctrl+Shift+'`		*Guides and Grid*
Show Guides	`Ctrl+;`		*Guides and Grid*
Snap to Guides	`Ctrl+Shift+;`		*Guides and Grid*
Smart Guides	`Ctrl+Alt+;`		
Manage Guides…			*The Guides Panel*
Clear Guides			*The Guides Panel*
Show Rulers	`Ctrl+R`		
Show Bounding Boxes	`Ctrl+Shift+B`		
Show Pages			
Įrankinės ►			*Skydeliai ir įrankinės*
Skydeliai ►			*Skydeliai ir įrankinės*
Toggle Full Screen	`F11`		*Vaizdas*
Perjungti skydelio matomumą	`Ctrl+Tab`		*Vaizdas*

Items menu

The *Items* helps you configure items' position in the layout and the relations between them (see *Interacting with layout items*).

18.4 Table : „Items“ Menu Tools

Įrankis	Kombinacija	Įrankinė	Aprašymas
Group	`Ctrl+G`	*Actions*	*Grouping items*
Ungroup	`Ctrl+Shift+G`	*Actions*	*Grouping items*
Raise	`Ctrl+]`	*Actions*	*Alignment and distribution*
Lower	`Ctrl+[`	*Actions*	*Alignment and distribution*
Bring to Front	`Ctrl+Shift+]`	*Actions*	*Alignment and distribution*
Send to Back	`Ctrl+Shift+[`	*Actions*	*Alignment and distribution*
Lock Selected Items	`Ctrl+L`	*Actions*	*Locking items*
Unlock All	`Ctrl+Shift+L`	*Actions*	*Locking items*
Align Items ►		*Actions*	*Alignment and distribution*
Distribute Items ►		*Actions*	*Moving and resizing items*
Resize ►		*Actions*	*Moving and resizing items*

Add Item menu

These are tools to create layout items. Each of them is deeply described in *Layout Items* chapter.

18.5 Table : „Add Item“ Menu Tools

Įrankis	Įrankinė	Aprašymas
Add Map	*Toolbox*	*The Map Item*
Add 3D Map	*Toolbox*	*The 3D Map Item*
Add Picture	*Toolbox*	*The Picture Item*
Add Label	*Toolbox*	*The Label Item*
Add Dynamic Text ►		*The Label Item*
Add Legend	*Toolbox*	*The Legend Item*
Add Scale Bar	*Toolbox*	*The Scale Bar Item*
Add North Arrow	*Toolbox*	*The North Arrow Item*
Add Shape ►	*Toolbox*	*The Regular Shape Item*
► *Add Rectangle*	*Toolbox*	*The Regular Shape Item*
► *Add Ellipse*	*Toolbox*	*The Regular Shape Item*
► *Add Triangle*	*Toolbox*	*The Regular Shape Item*
Add Marker	*Toolbox*	*The Marker Item*
Add Arrow	*Toolbox*	*The Arrow Item*
Add Node Item ►	*Toolbox*	*The Node-Based Shape Items*
► *Add Polygon*	*Toolbox*	*The Node-Based Shape Items*
► *Add Polyline*	*Toolbox*	*The Node-Based Shape Items*
Add HTML	*Toolbox*	*The HTML Frame Item*
Add Attribute Table	*Toolbox*	*The attribute table item*
Add Fixed Table	*Toolbox*	*The fixed table item*
Add Elevation Profile	*Toolbox*	*The Elevation Profile Item*

Atlas menu

18.6 Table : „Atlas“ Menu Tools

Įrankis	Kombinacija	Įrankinė	Aprašymas
Preview Atlas	`Ctrl+ALt+/`	*Atlas*	*Preview and generate an atlas*
First Feature	`Ctrl+<`	*Atlas*	*Preview and generate an atlas*
Previous Feature	`Ctrl+,`	*Atlas*	*Preview and generate an atlas*
Next Feature	`Ctrl+.`	*Atlas*	*Preview and generate an atlas*
Last feature	`Ctrl+>`	*Atlas*	*Preview and generate an atlas*
Print Atlas…		*Atlas*	*Preview and generate an atlas*

continues on next page

Table 18.6 – tęsinys iš praeito puslapio

Įrankis	Kombinacija	Įrankinė	Aprašymas
Export Atlas as Images…		*Atlas*	*Preview and generate an atlas*
Export Atlas as SVG…		*Atlas*	*Preview and generate an atlas*
Export Atlas as PDF…		*Atlas*	*Preview and generate an atlas*
Atlas Settings		*Atlas*	*Generate an Atlas*

Settings Menu

The *Settings* ► *Layout Options…* menu is a shortcut to *Settings* ► *Options* ► *Layouts* menu of QGIS main canvas. Here, you can set some options that will be used as default on any new print layout:

- *Layout defaults* let you specify the default font to use;
- With *Grid appearance*, you can set the grid style and its color. There are three types of grid: **Dots**, **Solid** lines and **Crosses**;
- *Grid and guide defaults* defines spacing, offset and tolerance of the grid (see *Guides and Grid* for more details);
- *Layout Paths*: to manage list of custom paths to search print templates.

The *Settings* ► *Keyboard Shortcuts…* menu allows you to use the *shortcuts manager* in the print layout interface.

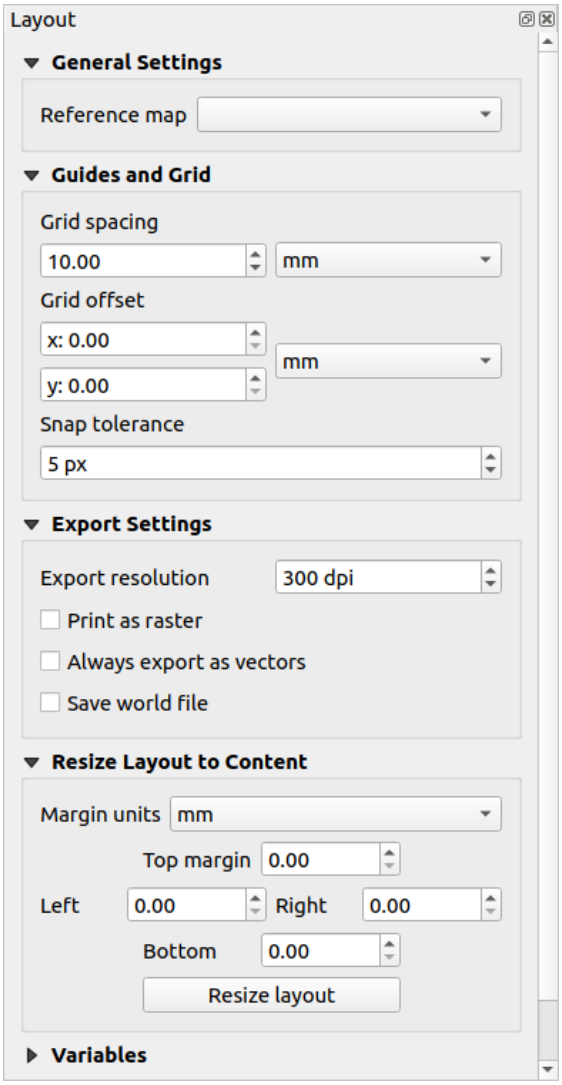
Contextual menus

Depending on where you right-click in the print layout dialog, you open a contextual menu with various features:

- Right-click on the menu bar or any toolbar and you get the list of layout panels and toolbars you can enable or disable in one-click.
- Right-click over a ruler and you can *Show Guides*, *Snap to Guides*, *Manage Guides…* opening the *Guides panel* or *Clear Guides*. It's also possible to hide the rulers.
- Right-click in the print layout canvas and:
  - You'll be able to *Undo* and *Redo* recent changes, or *Paste* any copied item (only available if no item is selected).
  - If you click over a page, you can additionally access the current *Page Properties* panel or *Remove Page*.
  - If you click on a selected item then you can cut or copy it as well as open the *Item Properties* panel.
  - If more than one item are selected, then you can either group them and/or ungroup if at least one group is already in the selection.
- Right-click inside a text box or spinbox widget of any layout panel provides edit options to manipulate its content.

The Layout Panel

In the *Layout* panel, you can define the global settings of your print layout.



18.3 Fig. : Layout Settings in the Print Layout

General settings

In a print layout, you can use more than one map item. The *Reference map* represents the map item to use as the layout's master map. It's assigned as long as there's a map item in the layout. The layout will use this map in any of their properties and variables calculating units or scale. This includes exporting the print layout to georeferenced formats.

Moreover, new layout items such as scale bar, legend or north arrow have by default their settings (orientation, displayed layers, scale, …) bound to the map item they are drawn over, and fall back to the reference map if there is no overlapping map.

Guides and Grid

You can put some reference marks on your paper sheet to help you accurately place some items. These marks can be:

- simple horizontal or vertical lines (called **Guides**) put at the position you want (see *The Guides Panel* for guides creation).
- or regular **Grid**: a network of horizontal and vertical lines superimposed over the layout.

Settings like *Grid spacing* or *Grid offset* can be adjusted in this group as well as the *Snap tolerance* to use for items. The tolerance is the maximum distance below which the mouse cursor is snapped to a grid or a guide, while moving, resizing or creating an item.

Whether grid or guides should be shown is set in *View* menu. There, you can also decide if they might be used to snap layout items. When both a grid line and a guide line are within tolerance of a point, guides will always take precedence - since they have been manually set (hence, assumption that they have been explicitly placed at highly desirable snapping locations, and should be selected over the general grid).

Pastaba

In the *Settings* ► *Layout Options* menu, you can also set the grid and guides parameters exposed above. However, these options will only apply as defaults to new print layouts.

Eksporto nustatymai

You can define a resolution to use for all exported maps in *Export resolution*. This setting can then be overridden each time you export a map.

Because of some advanced rendering options (*blending mode*, *effects*…), a layout item may need rasterization in order to be exported correctly. QGIS will individually rasterize it without forcing every other item to also be rasterized. This allows printing or saving as PostScript or PDF to keep items as much as possible as vectors, e.g. a map item with layer opacity won't force labels, scale bars, etc to be rasterized too. You can however:

- force all the items to be rasterized checking the *Print as raster* box;
- or use the opposite option, i.e. *Always export as vectors*, to force the export to keep items as vectors when exported to a compatible format. Note that in some cases, this could cause the output to look different to layout.

Where the format makes it possible (e.g., `.TIF`, `.PDF`) exporting a print layout results by default in a georeferenced file (based on the *Reference map* item in the *General settings* group). For other formats, georeferenced output requires you to generate a world file by checking *Save world file*. The world file is created beside the exported map(s), has the name of the page output with the reference map item and contains information to georeference it easily.

Resize layout to content

Using the *Resize page* tool in this group, you create a unique page composition whose extent covers the current contents of the print layout (with some optional *margins* around the cropped bounds).

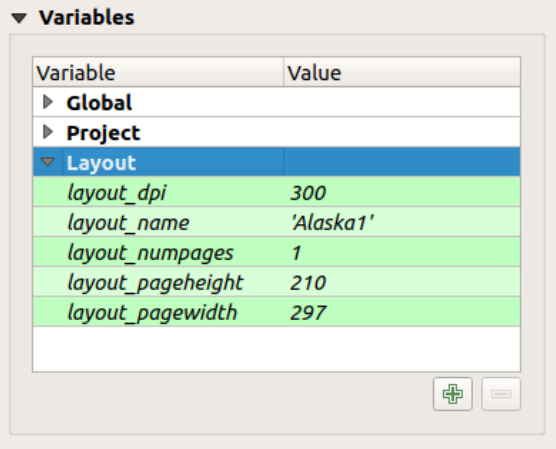
Note that this behavior is different from the *crop to content* option in that all the items are placed on a real and unique page in replacement of all the existing pages.

Variables

The *Variables* lists all the variables available at the layout's level (which includes all global and project's variables).

It also allows the user to manage layout-level variables. Click the button to add a new custom layout-level variable. Likewise, select a custom layout-level variable from the list and click the button to remove it.

More information on variables usage in the *General Tools* section.



18.4 Fig. : Variables Editor in the Print Layout

Working with the page properties

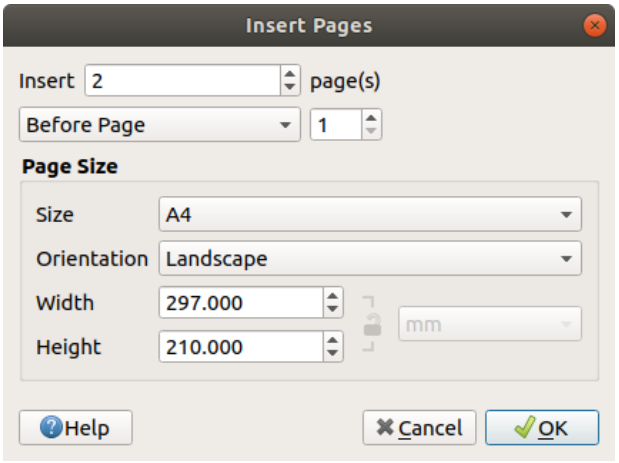
A layout can be composed of several pages. For instance, a first page can show a map canvas, and a second page can show the attribute table associated with a layer, while a third one shows an HTML frame linking to your organization website. Or you can add many types of items on each page.

Adding a new page

Futhermore, a layout can be made using different size and/or orientation of pages. To add a page, select the *Add Pages…* tool from the *Layout* menu or *Layout Toolbar*. The *Insert Pages* dialog opens and you are asked to fill:

- the number of pages to insert;
- the position of the page(s): before or after a given page or at the end of the print layout;
- The *Page size*: it could be of a preset format page (`A4`, `B0`, `Legal`, `Letter`, `ANSI A`, `Arch A` and their derivatives as well as a resolution type, such as `1920x1080` or `1024x768`) with associated *Orientation* (Portrait or Landscape).

  The page size can also be of a `custom` format; In that case, you'd need to enter its *Width* and *Height* (with locked size ratio if needed) and select the unit to use among `mm`, `cm`, `px`, `pt`, `in`, `ft`… Conversion of entered values is automatically applied when switching from one unit to another.



18.5 Fig. : Creating a new page in the Print Layout

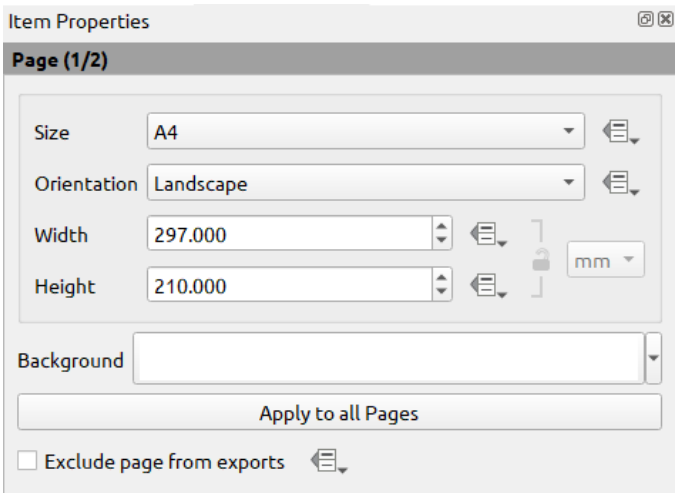
Updating page properties

Any page can be later customized through the Page *Item Properties* panel. To access a page's properties, either:

- Left-click on an empty section of the page
- Right-click on a page and select *Page Properties...*.
- Go to *Layout* ► *Page properties...* while hovering over the page

The *Item Properties* panel opens with settings such as:

- for layouts with several pages, it displays the active page number and the count of available pages
- the *Page size* frame described above. You can modify each property using the data defined override options (see *Explore Data-defined override buttons with atlas* for a use case);
- the *Background* of the current page using the *color* or *symbol* you want.
- the *Apply to all Pages* button updates the current page properties to the other existing pages in the layout;
- the *Exclude page from exports* to control whether the current page with its content should be included in the *layout output*;

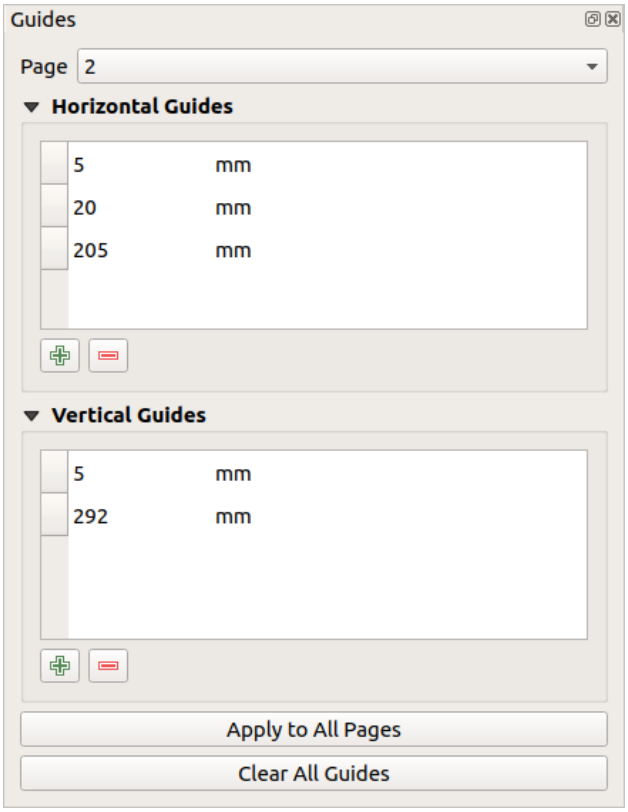


18.6 Fig. : Page properties dialog

The Guides Panel

Guides are vertical or horizontal line references you can place on a layout page to assist you on items placement, when creating, moving or resizing them. To be active, guides require the *View* ► *Show Guides* and *View* ► *Snap to Guides* options to be checked. To create a guide, there are two different methods:

- if the *View* ► *Show Rulers* option is set, drag out a ruler and release the mouse button within the page area, at the desired position.
- for more precision, use the *Guides* panel from the *View* ► *Toolbox* ► or by selecting *Manage guides for page…* from the page's contextual menu.



18.7 Fig. : The Guides panel

The *Guides* panel allows creation of snap lines at specific locations:

1. Select the *Page* you'd like to add the guides to
2. Click the Add new guide button and enter the coordinates of the horizontal or vertical line. The origin is at the top left corner. Different units are available for this.

   The panel also allows adjusting the position of existing guides to exact coordinates: double-click and replace the value.
3. The *Guides* panel lists only the items for the current page. It allows creation or removal of guides only in the current page. However, you can use the *Apply to All Pages* button to replicate the guide configuration of the current page to the other pages in the layout.
4. To delete a guide, select it and press the Remove selected guide button. Use *Clear All Guides* to remove all the guides in the current page.

Patarimas

Snapping to existing layout items

Other than guides and grids, you can use existing items as snapping references when moving, resizing or creating new items; these are called **smart guides** and require *View* ► *Smart Guides* option to be checked. Anytime the mouse pointer is close to an item's bound, a snapping cross appears.

The Items Panel

The *Items* panel offers some options to manage selection and visibility of items. All the items added to the print layout canvas (including *items group*) are shown in a list and selecting an item makes the corresponding row selected in the list as well as selecting a row does select the corresponding item in the print layout canvas. This is thus a handy way to select an item placed behind another one. Note that a selected row is shown as bold. Multiple items selection is also possible holding `Shift` for contiguous items, and `Ctrl` key for non-contiguous ones.

For any available item, you can:

- set it visible or not;
- lock or unlock its position;
- sort its Z position. You can move up and down each item in the list with a click and drag. The upper item in the list will be brought to the foreground in the print layout canvas. By default, a newly created item is placed in the foreground.
- change the item ID by double-clicking the text;
- right-click an item and select whether to copy or delete it or open its *properties panel*.

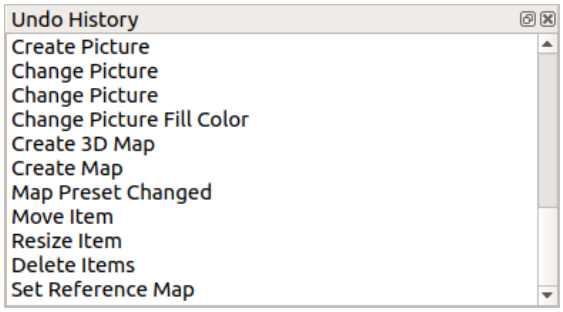
Once you have found the correct position for an item, you can lock it by ticking the box in column. Locked items are **not** selectable on the canvas. Locked items can be unlocked by selecting the item in the *Items* panel and unchecking the tickbox or you can use the icons on the toolbar.

The Undo History Panel: Revert and Restore actions

During the layout process, it is possible to revert and restore changes. This can be done with the revert and restore tools available in the *Edit* menu, the *Layout* toolbar or the contextual menu any time you right-click in the print layout area:

- Revert last change
- Restore last change

This can also be done by mouse click within the *Undo history* panel (see Fig. 18.8). The History panel lists the last actions done within the print layout. Select the point you want to revert to and once you do a new action all the actions done after the selected one will be removed.



18.8 Fig. : Undo History in the Print Layout

18.2 Layout Items

18.2.1 Layout Items Common Options

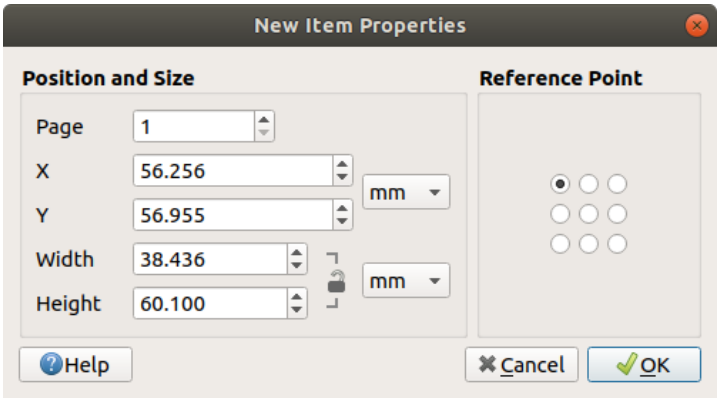
QGIS provides a large set of items to layout a map. They can be of map, legend, scale bar, picture, table, north arrow, image type… They however share some common options and behavior that are exposed below.

Creating a layout item

Items can be created using different tools, either from scratch or based on existing items.

To create a layout item from scratch:

1. Select the corresponding tool either from the *Add Item* menu or the *Toolbox* bar.
2. Then:
   - Click on the page and fill the size and placement information requested in the *New Item Properties* dialog that pops up (for details, see *Position and Size*);



18.9 Fig. : New Item properties dialog

   - Or click-and-drag to define the initial size and placement of the item. You can rely on *grids and guides* snapping for a better position.

Pastaba

Because they can have particular shapes, drawing node or arrow items does not work with one-click nor click-and-drag methods; you need to click and place each node of the item. See *The Node-Based Shape Items* for more details.

You can also:

1. Select an existing item with the Select/Move item button from the *Toolbox* toolbar
2. Use the contextual menu or the *Edit* menu tools to copy/cut the item and paste it at the mouse position as a new item.

   You can also use the *Paste in Place* (`Ctrl+Shift+V`) command to duplicate an item from one page to another and place it in the new page at the same coordinates as the original.

Moreover, you can create items using a print layout template (for details, see *The Layout Manager*) through the *Layout* ► *Add Items from Template…* command.

Patarimas

Add layout items using the file browser

From your file browser or using the *Browser* panel, drag-and-drop a print layout template (`.qpt` file) onto a print layout dialog and QGIS automatically adds all items from that template to the layout.

Interacting with layout items

Each item inside the print layout can be moved and resized to create a perfect layout. For both operations the first step is to activate the Select/Move item tool and click on the item.

You can select multiple items with the Select/Move item button: click and drag over the items or hold the `Shift` button and click on each of the items you want. To deselect an item, click on it holding the `Shift` button.

Each time there's a selection, count of selected items is displayed on the status bar. Inside the *Edit* menu, you can find actions to select all the items, clear all selections, invert the current selection and more…

Moving and resizing items

Unless *View* ► *Show Bounding Boxes* option is unchecked, a selected item will show squares on its boundaries; moving one of them with the mouse will resize the item in the corresponding direction. While resizing, holding `Shift` will maintain the aspect ratio. Holding `Alt` will resize from the item center.

To move a layout item, select it with the mouse and move while holding the left button. If you need to constrain the movements to the horizontal or vertical axis, hold the `Shift` button on the keyboard while moving the mouse. You can also move a selected item using the `Arrow keys` on the keyboard; if the movement is too slow, you can speed it up by holding `Shift`. If you need better precision, use the *Position and size* properties, or grid/guides snapping as explained above for item's creation.

Resizing or moving several items at once is made the same way as for a single item. QGIS however provides some advanced tools to automatically resize a selection of items following different rules:

- each item height matches the tallest or the shortest selected item;
- each item width matches the widest or the narrowest selected item;
- resizes items to squares: each item is enlarged to shape a square.

Likewise, there are *tools* available to organize multiple items position by distributing equidistantly:

- edges (left, right, top or bottom) of items;
- centers of items horizontally or vertically;
- spacing between items horizontally or vertically.

Grouping items

Grouping items allows you to manipulate a set of items like a single one: you can easily resize, move, delete, copy the items as a whole.

To create a group of items, select more than one and press the . *Group* button on the *Items* menu or the *Actions* toolbar, or from the right-click menu. A row named `Group` is added to the *Items* panel and can be locked or hidden like any other *Items panel's object*. Grouped items are **not individually** selectable on the canvas; use the Items panel for direct selection and access the item's properties panel.

Locking items

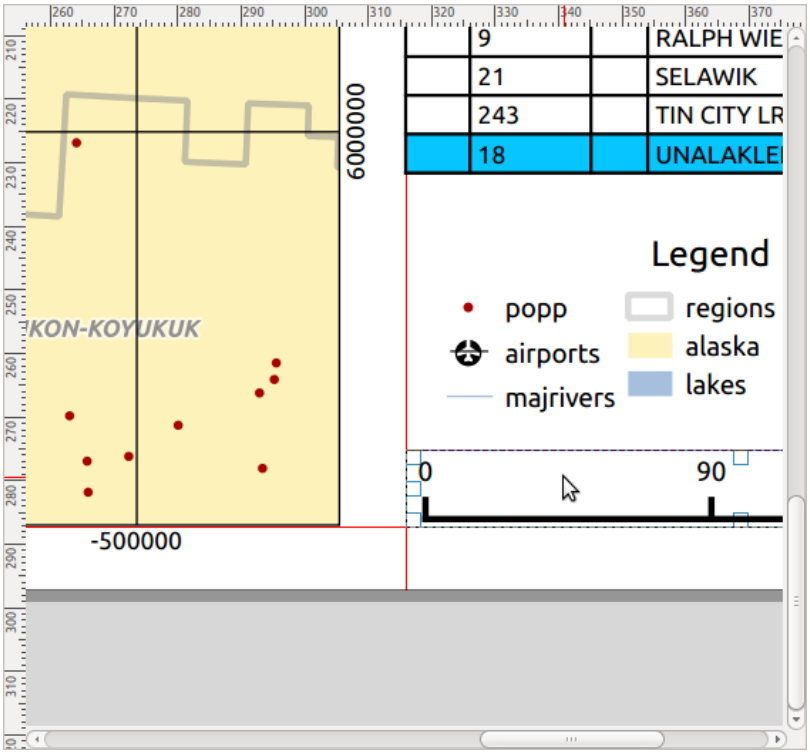
Once you have found the correct position for an item, you can lock it by using the *Lock selected items* button in the *Items* menu or the *Actions* toolbar, or ticking the box next to the item in the *Items* panel. Locked items are **not** selectable on the canvas.

Locked items can be unlocked by selecting the item in the *Items* panel and unchecking the tickbox or you can use the icons on the toolbar.

Alignment and distribution

Tools for raising or lowering the Z position of items in the layout are inside the

pull-down menu. Choose an element on the print layout canvas and select the matching functionality to raise or lower the selected element over the other elements. This order is shown in the *Items* panel. You can also raise or lower objects in the *Items* panel by clicking and dragging an object's label in this list.



18.10 Fig. : Alignment helper lines in the print layout

There are several alignment options available within the Align selected items pull-down menu (see Fig. 18.10). To use an alignment function, you first select the elements and then click on one of the alignment icons:

- *Align Left* or *Align Right*;
- *Align Top* or *Align Bottom*;
- *Align Center* horizontally or *Align Center Vertical*.

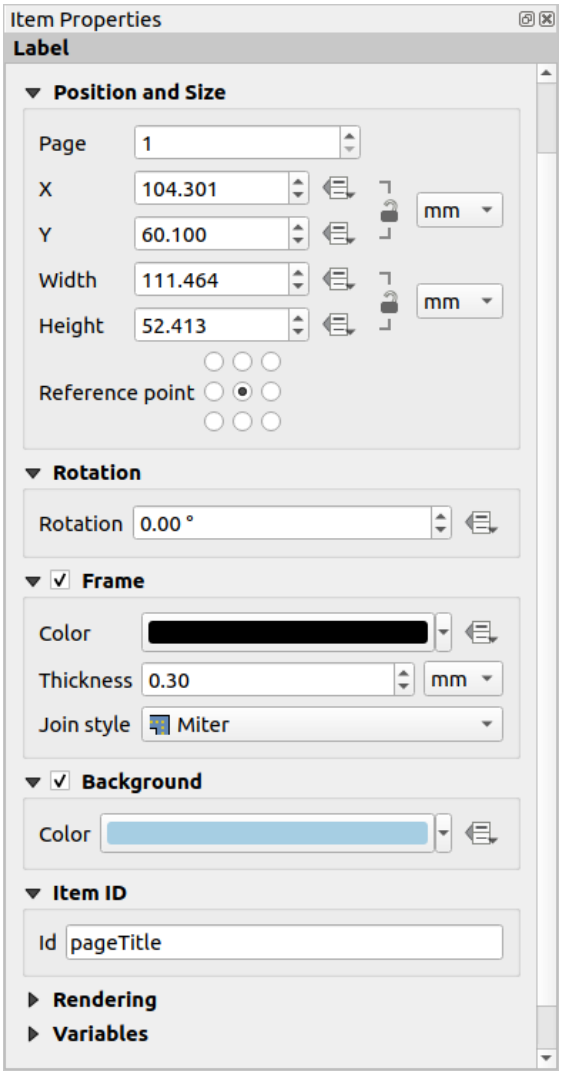
All selected elements will then be aligned to their common bounding box. When moving items on the layout canvas, alignment helper lines appear when borders, centers or corners are aligned.

Another way to improve layout items placement is by adjusting the spacing between them, over the layout page. This can be done by selecting the items and press the *Distribute Left Edges* drop-down menu to:

- *Distribute Left Edges* or *Distribute Right Edges* of items equidistantly
- *Distribute Top edges* or *Distribute Bottom Edges* of items equidistantly
- *Distribute Horizontal Centers* or *Distribute Vertical Centers* of items equidistantly
- Add equal space between items: *Distribute Horizontal Spacing Equally* or *Distribute Vertical Spacing Equally*

Items Common Properties

Layout items have a set of common properties you will find at the bottom of the *Item Properties* panel: Position and size, Rotation, Frame, Background, Item ID, Variables and Rendering (see Fig. 18.11).



18.11 Fig. : Common Item Properties groups

Pastaba

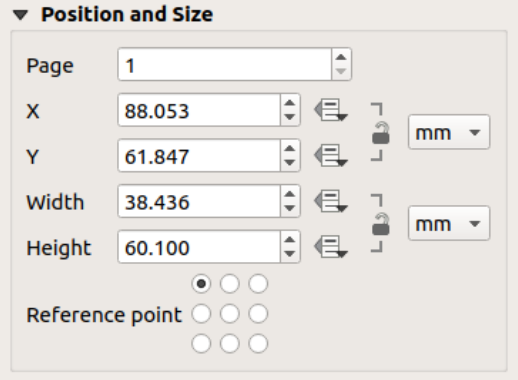
The Data defined override icon next to most of the options means that you can associate that property with a layer, features attributes, geometry or with any other layout item's property, using *expressions* or *variables*. For more information see *Nuo duomenų priklausančių permušimų nustatymai*.

- The *Position and size* group lets you define the size and position of the frame which contains the item (see *Position and Size* for more information).
- The *Rotation* sets the rotation of the element (in degrees).
- The *Frame* shows or hides the frame around the item. Use the *Color*, *Thickness* and *Join style* widgets to adjust those properties.

- In the *Background* group you can create or pick a background *Color*. Transparency can be adjusted through altering the alpha field settings.
- Use the *Item ID* to create a relationship to other print layout items. This is used with QGIS server and other potential web clients. You can set an ID on an item (for example, a map or a label), and then the web client can send data to set a property (e.g., label text) for that specific item. The GetProjectSettings command will list the items and IDs which are available in a layout.
- *Rendering* mode helps you set whether and how the item can be displayed: you can, for instance, apply *blending mode*, adjust the opacity of the item or *Exclude item from exports*.

Position and Size

Extending the features of the *New Item Properties* dialog with data-defined capabilities, this group allows you to place the items accurately.

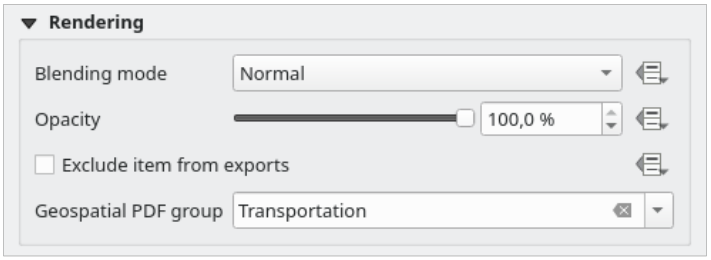


18.12 Fig. : Position and size

- the actual number of the page to place the item on;
- the reference point of the item;
- the *X* and *Y* coordinates of the *Reference point* of the item on the chosen page. The ratio between these values can be locked by clicking on the button. Changes made to a value using the widget or the Select/Move item tool will be reflected in both of them;
- the *Width* and *Height* of the item bounding box. As for coordinates, the ratio between width and height can be locked.

Rendering mode

QGIS allows advanced rendering for layout items just like vector and raster layers.



18.13 Fig. : Rendering mode

- *Blending mode*: With this tool you can achieve effects which would otherwise only be achieved using graphic rendering software. The pixels of your overlaying and underlaying items can be mixed according to the mode set (see *Suliejimo režimai* for description of each effect).
- *Opacity*: You can make the underlying item in the layout visible with this tool. Use the slider to adapt the visibility of your item to your needs. You can also make a precise definition of the percentage of visibility in the menu beside the slider.
- *Exclude item from exports*: Combined with data-defined expressions, you can decide to make an item invisible in some exports. After activating this checkbox, the item will not be included in export to PDF, print etc..
- *Geospatial PDF group*: When set, a matching layer tree group will be created in the exported Geospatial PDF and the item will only be visible when this group is checked. This allows content to be selectively displayed as a group by viewers of the Geospatial PDF. E.g., it can allow extra layout content such as descriptive labels or legends to only be shown when layers from the group are visible, making the file export much more flexible.

Variables

The *Variables* lists all the variables available at the layout item's level (which includes all global, project and layout's variables). Layout map items alhave an additional *Map settings* section for variables that provide easy access to values like the map's scale, extent, and so on.

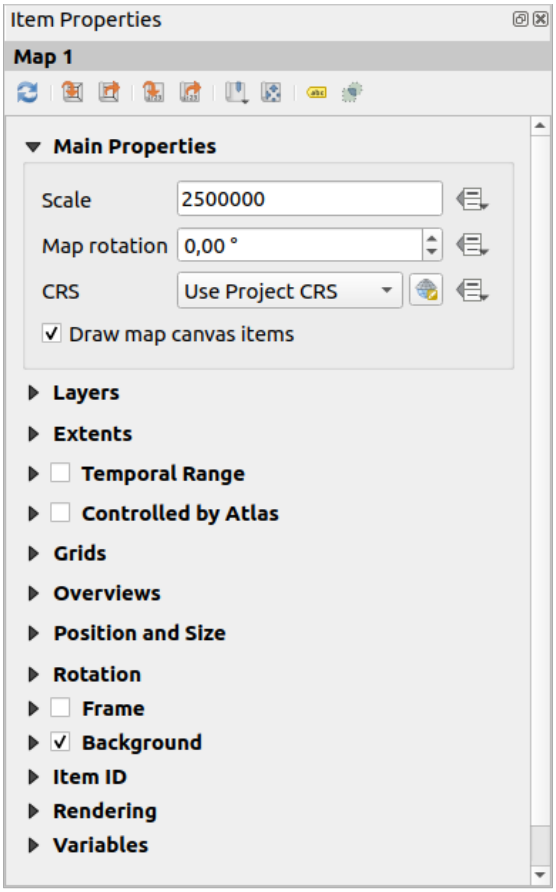
In *Variables*, it's also possible to manage layout item level variables. Click the button to add a new custom variable. Likewise, select any custom item-level variable from the list and click the button to remove it.

More information on variables usage in the *Reikšmių saugojimas kintamuosiuose* section.

18.2.2 The Map Item

The map item is the main frame that displays the map you've designed in the map canvas. Use the *Add Map* tool following *items creation instructions* to add a new map item that you can later manipulate the same way as exposed in *Interacting with layout items*.

By default, a new map item shows the current status of the *map canvas* with its extent and visible layers. You can customize it thanks to the *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities:



18.14 Fig. : Map Item Properties Panel

The Toolbar

The Map *Item Properties* panel embeds a toolbar with the following functionalities:

- Update map preview
- Set map canvas to match main canvas extent
- View current map extent in main canvas
- Set map scale to match main canvas scale
- Set main canvas to match current map scale
- Bookmarks: set the map item extent to match an existing spatial bookmark
- Interactively edit map extent: pan and zoom interactively within the map item
- Labeling settings: control feature label behaviour (placement, visibility…) in the layout map item extent:
  - set a *Margin from map edges*, a data definable distance from the map item's limits inside which no label should be displayed
  - *Allow truncated labels on edges of map*: controls whether labels which fall partially outside of the map item allowed extent should be rendered. If checked, these labels will be shown (when there's no way to place them fully within the visible area). If unchecked then partially visible labels will be skipped.

  - *Label blocking items*: allows other layout items (such as scalebars, north arrows, inset maps, etc) to be marked as a blockers for the map labels in the **active** map item. This prevents any map labels from being placed under those items - causing the labeling engine to either try alternative placement for these labels or discard them altogether.

    If a *Margin from map edges* is set, the map labels are not placed closer than the specified distance from the checked layout items.

  - *Show unplaced labels*: can be used to determine whether labels are missing from the layout map (e.g. due to conflicts with other map labels or due to insufficient space to place the label) by highlighting them in a *predefined color*.

- Clipping settings: allows to clip the map item to the atlas feature and to shape and polygon items:

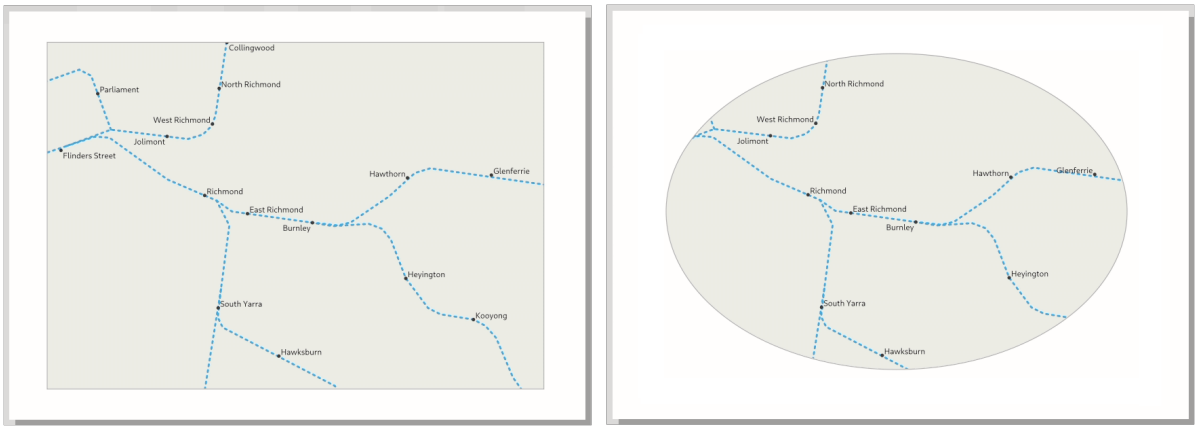
  - *Clip to atlas feature*: you can determine that the layout map item will be clipped automatically to the current *atlas feature*.

    There are different clipping modes available:

    * *Clip During Render Only*: applies a painter based clip, so that portions of vector features which sit outside the atlas feature become invisible
    * *Clip Feature Before Render*: applies the clip before rendering features, so borders of features which fall partially outside the atlas feature will still be visible on the boundary of the atlas feature
    * *Render Intersecting Features Unchanged*: renders all features which intersect the current atlas feature, but without clipping their their geometry.

    You can *Force labels inside atlas feature*. If you don't want to *Clip all layers* to the atlas feature you can use the *Clip selected layers* option.

  - *Clip to item*: it is possible to change the shape of the map item by using a *shape* or *polygon* item from the print layout. When you enable this option the map will be automatically clipped to the selected shape in the combobox. Again, the above mentioned clipping modes are available and labels can be forced to display only inside the clipping shape.



18.15 Fig. : Clipping a layout map item to shapes

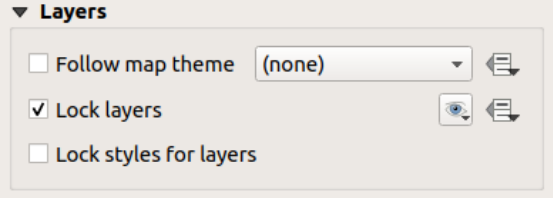
Main properties

In the *Main properties* group (see Fig. 18.14) of the map *Item Properties* panel, available options are:

- The *Update Preview* button to refresh the map item rendering if the view in map canvas has been modified. Note that most of the time, the map item refresh is automatically triggered by the changes;
- The *Scale* to manually set the map item scale;
- The *Map rotation* allows you to rotate the map item content clockwise in degrees. The rotation of the map canvas can be imitated here;
- The *CRS* allows you to display the map item content in any *CRS*. It defaults to `Use project CRS`;
- *Draw map canvas items* lets you show in the print layout *annotations* that are placed on the main map canvas.

Layers

By default, map item appearance is synced with the map canvas rendering meaning that toggling visibility of the layers or modifying their style in the *Layers Panel* is automatically applied to the map item. Because, like any other item, you may want to add multiple map items to a print layout, there's a need to break this synchronization in order to allow showing different areas, layer combinations, at different scales… The *Layers* properties group (see Fig. 18.16) helps you do that.



18.16 Fig. : Map Layers group

If you want to keep the map item consistent with an existing *map theme*, check *Follow map theme* and select the desired theme in the drop-down list. Any changes applied to the theme in QGIS' main window (using the replace theme function) will automatically affect the map item. If a map theme is selected, the *Lock styles for layers* option is disabled because *Follow map theme* also updates the style (symbology, labels, diagrams) of the layers.

To lock the layers shown in a map item to the current map canvas visibility, check *Lock layers*. When this option is enabled, any changes on the layers' visibility in QGIS' main window will not affect the layout's map item. Nevertheless, style and labels of locked layers are still refreshed according to QGIS' main window. You can prevent this by using *Lock styles for layers*.

Instead of using the current map canvas, you can also lock the layers of the map item to those of an existing map theme: select a map theme from the Set layer list from a map theme drop-down button, and the *Lock layers* is activated. The set of visible layers in the map theme is from now on used for the map item until you select another map theme or uncheck the *Lock layers* option. You then may need to refresh the view using the Refresh view button of the *Navigation* toolbar or the *Update Preview* button seen above.

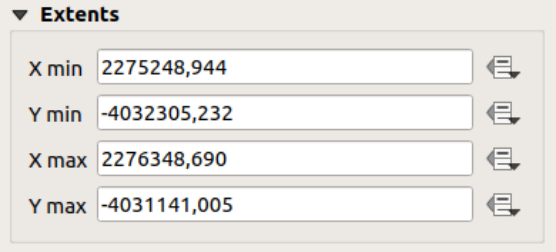
Note that, unlike the *Follow map theme* option, if the *Lock layers* option is enabled and set to a map theme, the layers in the map item will not be refreshed even if the map theme is updated (using the replace theme function) in QGIS' main window.

Locked layers in the map item can also be *data-defined*, using the icon beside the option. When used, this overrides the selection set in the drop-down list. You need to pass a list of layers separated by `|` character. The following example locks the map item to use only layers `layer 1` and `layer 2`:

```
concat ('layer 1', '|', 'layer 2')
```

Extents

The *Extents* group of the map item properties panel provides the following functionalities (see Fig. 18.17):



18.17 Fig. : Map Extents group

The **Extents** area displays `X` and `Y` coordinates of the area shown in the map item. Each of these values can be manually replaced, modifying the map canvas area displayed and/or map item size. The extent can also be modified using tools at the top of the map item panel such as:

- Set map canvas to match main canvas extent
- Set map scale to match main canvas scale

You can also alter a map item extent using the Move item content tool: click-and-drag within the map item to modify its current view, keeping the same scale. With the tool enabled, use the mouse wheel to zoom in or out, modifying the scale of the shown map. Combine the movement with `Ctrl` key pressed to have a smaller zoom.

Elevation range

The *Elevation range* setting in the map item properties allows you to filter the content of certain layers based on a specific elevation range. While all layers remain visible, those that support elevation filtering (currently point clouds and raster DEMs) will have their data filtered. Only the parts of these layers that fall within the elevation range set by the *Lower* and *Upper* values will be displayed.

The *Elevation range* can be data-defined. This means you can set different elevation ranges for different features in the *atlas* or *report*.

Temporal range

The *Temporal range* group of the map item properties panel provides the options to control layers rendering in the map item based on a temporal range. Only layers whose temporal properties overlap with the time range set by the *Start* and *End* dates are displayed in the map item.

The associated data-defined widgets help make the time range dynamic, and allow outputting temporal *atlases*, i.e. automated maps with fixed spatial extent and whose contents vary based on time. For example, using as coverage layer a csv file with a start and end pair of fields and a number of rows representing date ranges, enable both the temporal range and control by atlas in the map item properties and hit atlas export.

Controlled by atlas

The *Controlled by atlas* group properties is available only if an *atlas* is active in the print layout. Check this option if you want the map item being ruled by the atlas; when iterating over the coverage layer, the map item extent is panned/zoomed to the atlas feature following:

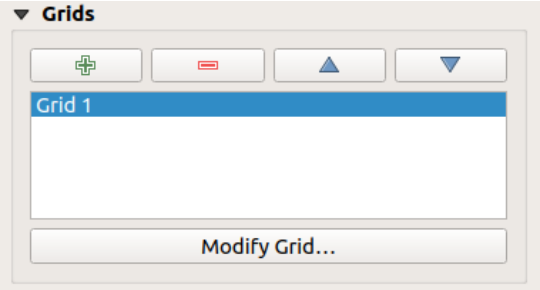
- *Margin around features*: zooms to the feature at the best scale, keeping around each a margin representing a percentage of the map item width or height. The margin can be the same for all features or *set variable*, e.g., depending on map scale;
- *Predefined scale (best fit)*: zooms to the feature at the project *predefined scale* where the atlas feature best fits;
- *Fixed scale*: atlas features are panned from one to another, keeping the same scale of the map item. Ideal when working with features of same size (e.g., a grid) or willing to highlight size differences among atlas features.

Grids

With grids, you can add, over your map, information relative to its extent or coordinates, either in the map item projection or a different one. The *Grids* group provides the possibility to add several grids to a map item.

- With the and buttons you can add or remove a selected grid;
- With the and buttons you can move up and down a grid in the list, hence move it on top or bottom of another one, over the map item.

Double-click the added grid to rename it.



18.18 Fig. : Map Grids Dialog

To modify a grid, select it and press the *Modify Grid…* button to open the *Map Grid Properties* panel and access its configuration options.

Grid Appearance

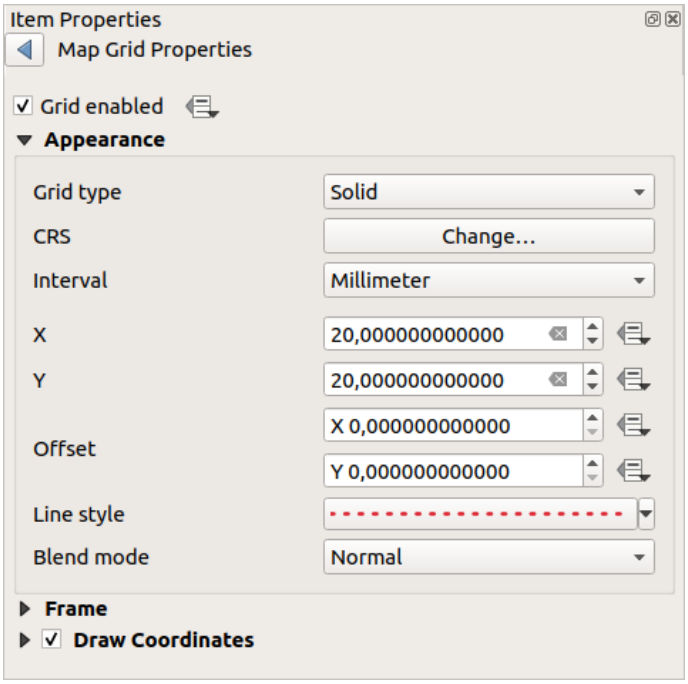
In the *Map Grid Properties* panel, check *Grid enabled* to show the grid on the map item.

As grid type, you can specify to use a:

- **Solid**: shows a line across the grid frame. The *Line style* can be customized using *color* and *symbol* selector widget;
- **Cross**: displays segment at the grid lines intersection for which you can set the *Line style* and the *Cross width*;
- **Markers**: only displays customizable markers symbol at grid lines intersection;
- or **Frame and annotations only**.

Other than the grid type, you can define:

- the *CRS* of the grid: by default, it will follow the map item CRS. Press Select CRS button to set it to a different CRS.
- the *Interval* type to use for the grid references:
  - **Map Units**: you set a distance within the map (in the unit of the grid CRS) between consecutive grid references in the *X* and *Y* directions. The number of grid ticks will vary depending on the map scale.
  - choosing **Fit Segment Width** will dynamically select the grid interval based on the map extent to a „pretty“ interval. That optimal interval is calculated within a range of distances whose *Minimum* and *Maximum* values can be customized.
  - With **Millimeters** or **Centimeters**, you set a distance on the paper between consecutive grid references in the *X* and *Y* directions. The number of grid ticks will be the same whatever the map scale.
- the *Offset* from the map item edges, in the *X* and/or the *Y* direction
- and the *Blend mode* of the grid (see *Suliejimo režimai*) when compatible.



18.19 Fig. : Grid Appearance Dialog

Grid Frame

There are different options to style the frame that holds the map. The following options are available: `No Frame`, `Zebra`, `Zebra (nautical)`, `Interior ticks`, `Exterior ticks`, `Interior and Exterior ticks`, `Line border` and `Line border (nautical)`.

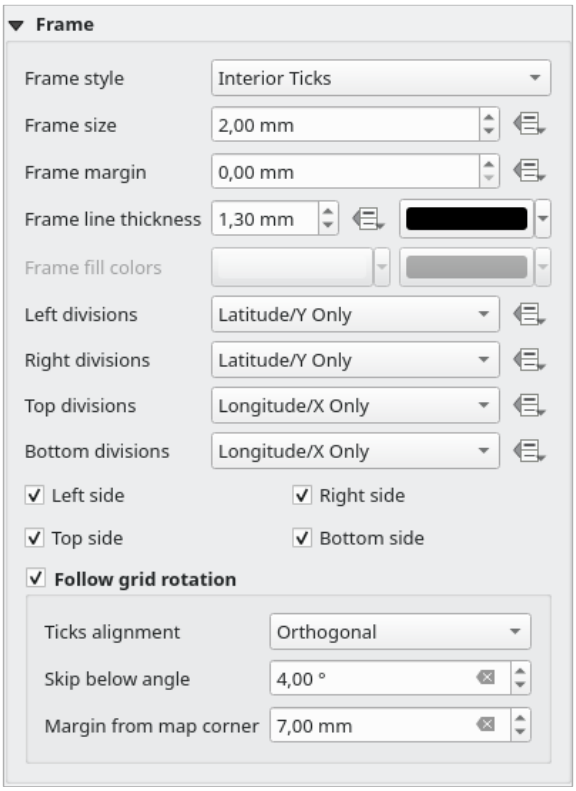
When compatible, it's possible to set the *Frame size*, a *Frame margin*, the *Frame line thickness* with associated color and the *Frame fill colors*.

Using `Latitude/Y only` and `Longitude/X only` values in the divisions section you can prevent a mix of latitude/Y and longitude/X coordinates showing on each side when working with rotated maps or reprojected grids. Also you can choose to set visible or not each side of the grid frame.

When the map item extent is rotated (from the *Main properties* group) or the grid has a different CRS applied, grid lines may not be orthogonal to the map item sides. This can result in bad looking of the grid when styled with interior

and/or exterior ticks. Checking *Follow grid rotation* will align the ticks with grid lines. Moreover, you can adjust some more properties:

- *Ticks alignment*: The interior and/or exterior ticks will be parallel to their corresponding grid line. Their alignment can be:
  - **Orthogonal**: ticks on the same side end at one line, parallel to the side. This can result e.g. in some ticks getting longer when with a low angle to the frame.
  - **Fixed length**: all ticks have the same length, so they may not align
- *Skip below angle*: prevents displaying ticks for grid lines intersecting the frame border below a specified threshold
- *Margin from map corner*: prevents displaying ticks too close to the map corners, because they could overlap and/or be out of bounds.



18.20 Fig. : Grid Frame Dialog

Coordinates

The *Draw coordinates* checkbox allows you to add coordinates to the map frame. Displayed values relate to the chosen *grid interval* unit. You can choose the annotation numeric format, the options range from decimal to degrees, minute and seconds, with or without suffix, aligned or not and a custom format using the expression dialog.

For each of the *Left*, *Right*, *Top* and *Bottom* sides of the grid frame, you can indicate:

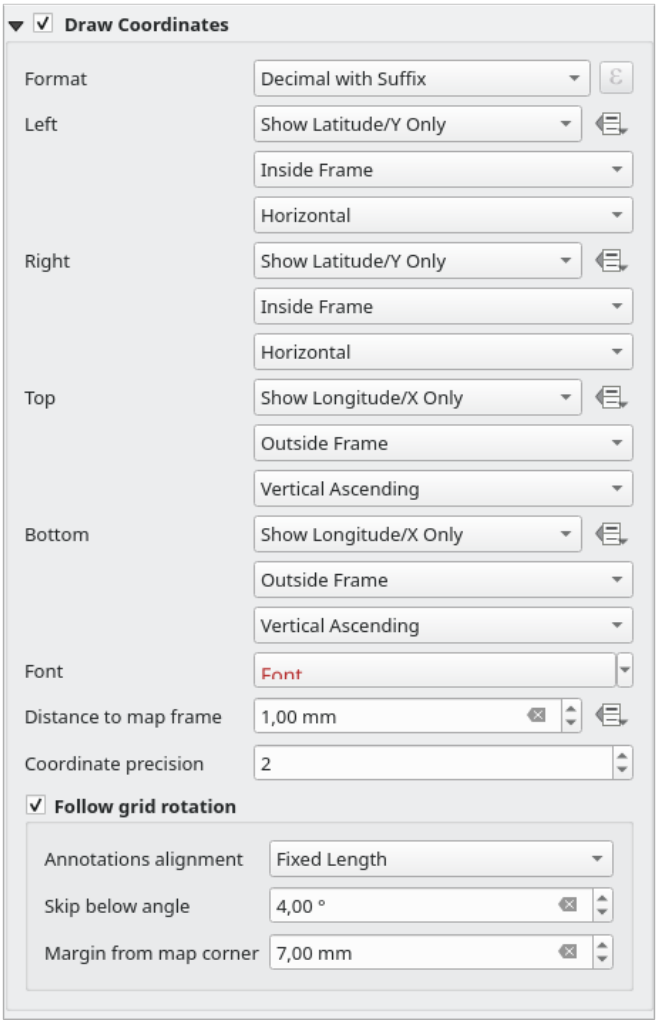
- whether to render the coordinates: **Show all**, **Show latitude/Y only**, **Show longitude/X only**, **Disabled**. Showing only Latitude/Y or Longitude/X values in the divisions helps prevent a mix of latitude/Y and longitude/X coordinates showing on each side when working with rotated maps or reprojected grids.
- the relative position of the text to the grid frame: **Outside frame** or **Inside frame**
- the placement and orientation of the annotation:

  - **Horizontal**
  - **Vertical ascending**, **Vertical descending**
  - **Boundary direction**
  - **Above tick**, **On tick**, **Under tick** when a tick-based frame is used

You can also define the *Font properties* (font, size, color, buffer,…) the *Distance to the map frame* and the *Coordinate precision* (number of decimals) for the drawn annotations.

Follow grid rotation: available when the map extent is rotated or the grid is reprojected, it helps you adjust the annotations placement. Depending on the selected placement mode, the annotations are also rotated:

- *Annotations alignment*: it can be **Orthogonal** or of **Fixed length**
- *Skip below angle*: prevents displaying annotations for grid lines intersecting the frame border below a specified threshold
- *Margin from map corner*: prevents displaying annotations too close to the map corners, because they could overlap and/or be out of bounds.

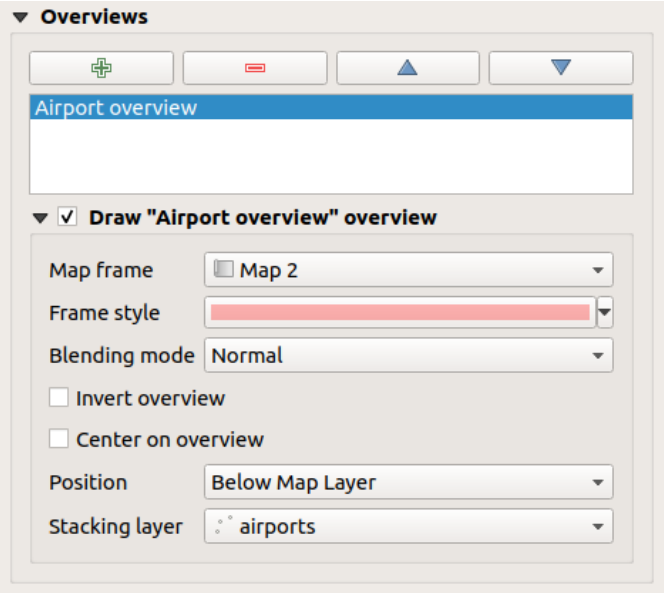


18.21 Fig. : Grid Draw Coordinates dialog

Overviews

Sometimes you may have more than one map in the print layout and would like to locate the study area of one map item on another one. This could be for example to help map readers identify the area in relation with its larger geographic context shown in the second map.

The *Overviews* group of the map panel helps you create the link between two different maps extent and provides the following functionalities:



18.22 Fig. : Map Overviews group

To create an overview, select the map item on which you want to show the other map item's extent and expand the *Overviews* option in the *Item Properties* panel. Then press the button to add an overview.

Initially this overview is named ‚Overview 1' (see Fig. 18.22). You can:

- Rename it with a double-click
- With the and buttons, add or remove overviews
- With the and buttons, move an overview up and down in the list, placing it above or below other overviews in the map item (when they are at the same *stack position*).

Then select the overview item in the list and check the *Draw „<name_overview>"overview* to enable the drawing of the overview on the selected map frame. You can customize it with:

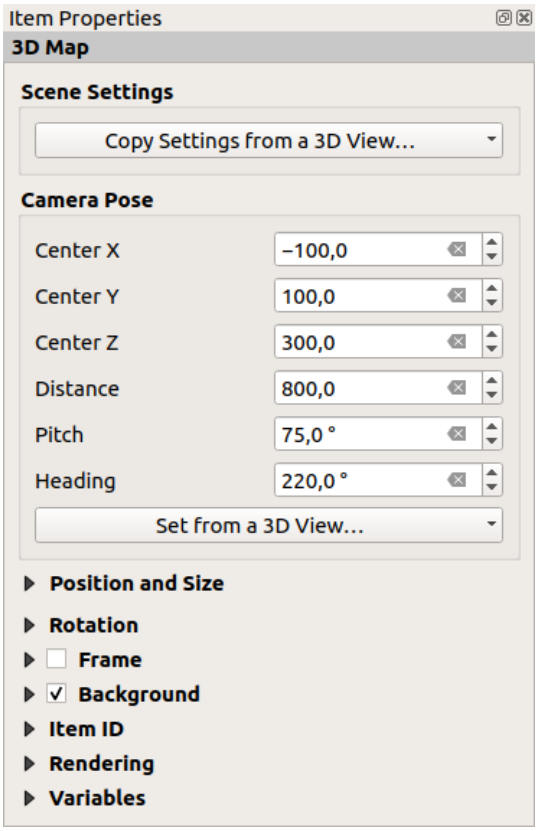
- The *Map frame* selects the map item whose extents will be shown on the present map item.
- The *Frame Style* uses the *symbol properties* to render the overview frame.
- The *Blending mode* allows you to set different transparency blend modes.
- The *Invert overview* creates a mask around the extents when activated: the referenced map extents are shown clearly, whereas the rest of the map item is blended with the frame fill color (if a fill color is used).
- The *Center on overview* pans the map item content so that the overview frame is displayed at the center of the map. You can only use one overview item to center, when you have several overviews.
- The *Position* controls exactly where in the map item's layer stack the overview will be placed, e.g. allowing an overview extent to be drawn below some feature layers such as roads whilst drawing it above other background layers. Available options are:

- *Below map*
- *Below map layer* and *Above map layer*: place the overview frame below and above the geometries of a layer, respectively. The layer is selected in the *Stacking layer* option.
- *Below map labels*: given that labels are always rendered above all the feature geometries in a map item, places the overview frame above all the geometries and below any label.
- *Above map labels*: places the overview frame above all the geometries and labels in the map item.

18.2.3 The 3D Map Item

The 3D Map item is used to display a *3D map view*. Use the *Add 3D Map* button, and follow *items creation instructions* to add a new 3D Map item that you can later manipulate the same way as demonstrated in *Interacting with layout items*.

By default, a new 3D Map item is empty. You can set the properties of the 3D view and customize it in the *Item Properties* panel. In addition to the *common properties*, this feature has the following functionalities (Fig. 18.23):



18.23 Fig. : 3D Map Item Properties

Scene settings

Press *Copy Settings from a 3D View…* to choose the 3D map view to display.

The 3D map view is rendered with its current configuration (layers, terrain, lights, camera position and angle…).

Camera pose

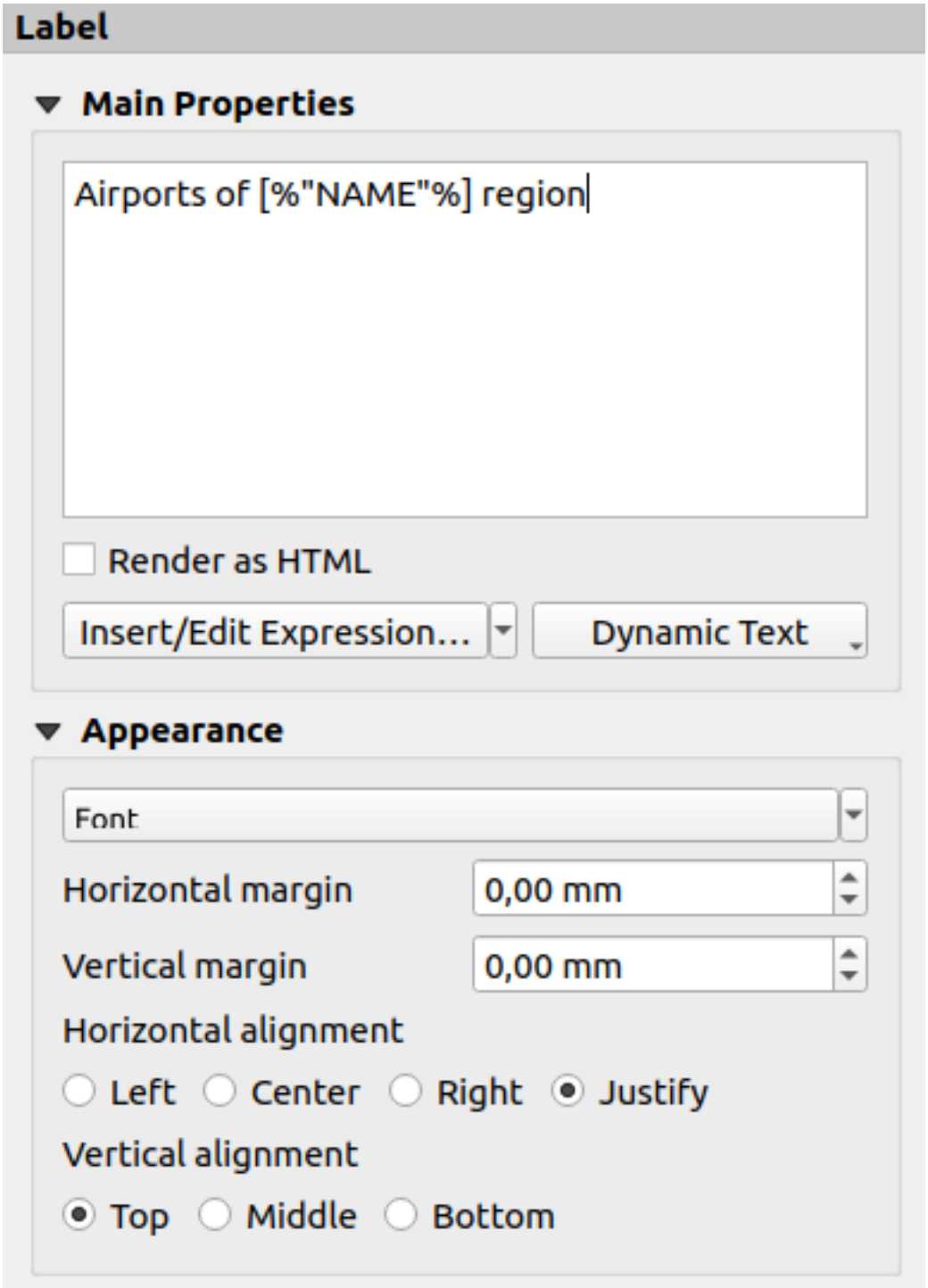
- *Center X* sets the X coordinate of the point the camera is pointing at
- *Center Y* sets the Y coordinate of the point the camera is pointing at
- *Center Z* sets the Z coordinate of the point the camera is pointing at
- *Distance* sets the distance from the camera center to the point the camera is pointing at
- *Pitch* sets the rotation of the camera around the X-axis (vertical rotation). Values from 0 to 360 (degrees). 0°: terrain seen straight from above; 90°: horizontal (from the side); 180°: straight from below; 270°: horizontal, upside down; 360°: straight from above.
- *Heading* sets the rotation of the camera around the Y-axis (horizontal rotation - 0 to 360 degrees). 0°/360°: north; 90°: west; 180°: south; 270°: east.

The *Set from a 3D View…* pull-down menu lets you populate the items with the parameters of a 3D View.

18.2.4 The Label Item

The *Label* item is a tool that helps decorate your map with texts that would help to understand it; it can be the title, author, data sources or any other information… You can add a label with the *Add Label* tool following *items creation instructions* and manipulate it the same way as exposed in *Interacting with layout items*.

By default, the label item provides a default text that you can customize using its *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities (see Fig. 18.24):



18.24 Fig. : Label Item Properties Panel

Main properties

The *Main properties* group is the place to provide the text of the label. The text can be static, dynamic with *expression* functions and variables, and/or formatted with HTML. Dynamic parts of a label need to be surrounded by `[%` and `%]` in order to be interpreted and evaluated as such.

- To use expressions in labels, you can click on *Insert/Edit Expression…* button, write your formula as usual and when the dialog is applied, QGIS automatically adds the surrounding characters.

  > **Patarimas**
  >
  > When you click the *Insert/Edit Expression…* button with no selection made in the textbox, the new expression will be appended to the existing text. However, if you want to modify an existing expression, the behavior changes based on your selection: If some text is selected and the selection is within an expression (between ‚[%' and ‚%]'), or if no text is selected but the cursor is inside an expression, the whole expression will be selected. In all other cases, the selection will remain as is.

  Because maps are usually filled with some common textual information (date, author, title, page number, …), QGIS provides a direct access to the corresponding expressions or variables: press the *Dynamic text* button to select and insert them into your label.

  > **Patarimas**
  >
  > The top menu *Add Item* ► *Add Dynamic Text* ► can be used to create a new label item filled with the selected predefined expression.

  It's possible to turn a dynamic label into static: press the drop-down arrow next to the *Insert/Edit Expression…* button and select *Convert to Static*. Any dynamic parts of the label's contents will be evaluated and replaced with their current values. You can then manually tweak the resulting text when needed.

- Labels can be interpreted as HTML code: check *Render as HTML*. You can now insert HTML tags or styles, URL, a clickable image that links to a web page, or something more complex…

The following code combines HTML rendering with expressions, for an advanced labeling and will output Fig. 18.25:

```
<html>
 <head>
   <style>
      /* Define some custom styles, with attribute-based size */
      name {color:red; font-size: [% ID %]px; font-family: Verdana; text-shadow:␣
↪grey 1px 0 10px;}
      use {color:blue;}
   </style>
 </head>

 <body>
   <!-- Information to display -->
   <u>Feature Information</u>
   <ul style="list-style-type:disc">
     <li>Feature Id: [% ID %]</li>
     <li>Airport: <name>[% NAME %]</name></li>
     <li>Main use: <use>[% USE %]</use></li>
   </ul>
   Last check: [% concat( format_date( "control_date", 'yyyy-MM-dd'), ' by <b><i>',
↪ @user_full_name, '</i></b>' ) %]

   <!-- Insert an image -->
   <p align=center><img src="path/to/logos/qgis-logo-made-with-color.svg" alt=
↪"QGIS icon" style="width:80px;height:50px;"</p>
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
 </body>
</html>
```

18.25 Fig. : Leveraging a label with HTML styling

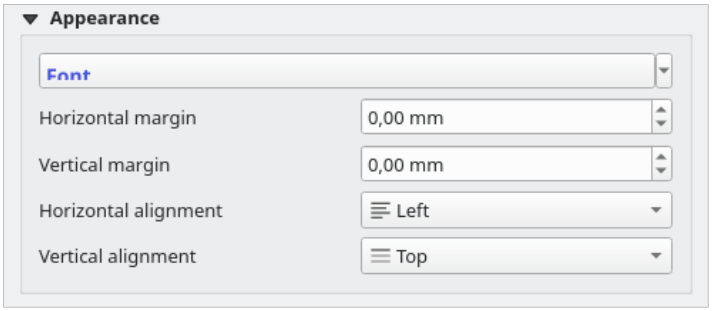
Appearance

- Define font and style of the text by clicking on the *Font* button. In the *Label Font* menu you can use some of the options for *Formatting the label text*.
- You can specify different horizontal and vertical margins in mm. This is the margin from the edge of the layout item. The label can be positioned outside the bounds of the label e.g. to align label items with other items. In this case you have to use negative values for the margin.
- Using the text alignment is another way to position your label. It can be:
  - *Left*, *Center*, *Right* or *Justify* for *Horizontal alignment*
  - and *Top*, *Vertical Center*, *Bottom* for *Vertical alignment*.

Patarimas

Resize the layout label item to fit its contents

By default, the size of a layout label item does not depend on the contained text. It may be too large with useless blank space or too small, hiding part of the text. To automatically resize a label so it fits the text inside it perfectly, double-click one of the square handles on the label. The label will adjust its size to match the text, either expanding or shrinking as needed, while the opposite handle of the label remains fixed in place. This allows you to quickly ensure the label fits the text without manually resizing it. So e.g., double-clicking the left-middle handle causes the left side of the label to move.



18.26 Fig. : Label Item Appearance Properties

Exploring expressions in a label item

Below some examples of expressions you can use to populate the label item with interesting information - remember that the code, or at least the calculated part, should be surrounded by `[%` and `%]` in the *Main properties* frame:

- Display a title with the current atlas feature value in „field1“:

```
'This is the map for ' || "field1"
```

  or, written in the *Main properties* section:

```
This is the map for [% "field1" %]
```

- Add a pagination for processed atlas features (eg, `Page 1/10`):

```
concat( 'Page ', @atlas_featurenumber, '/', @atlas_totalfeatures )
```

- Return the name of the airports of the current atlas region feature, based on their common attributes:

```
aggregate( layer := 'airports',
           aggregate := 'concatenate',
           expression := "NAME",
           filter := fk_regionId = attribute( @atlas_feature, 'ID' ),
           concatenator := ', '
         )
```

  Or, if an *attributes relation* is set:

```
relation_aggregate( relation := 'airports_in_region_relation',
                    aggregate := 'concatenate',
                    expression := "NAME",
                    concatenator := ', '
                  )
```

- Return the name of the airports contained in the current atlas region feature, based on their spatial relationship:

```
aggregate( layer := 'airports',
           aggregate := 'concatenate',
           expression := "NAME",
           filter := contains( geometry( @parent ), $geometry ),
           concatenator := ', '
         )
```

  OR:

```
array_to_string( array:= overlay_contains( layer := 'airports',
                                           expression := "NAME" ),
                 delimiter:= ', '
               )
```

- Return the lower X coordinate of the `Map 1` item's extent:

```
x_min( map_get( item_variables( 'Map 1' ), 'map_extent' ) )
```

- Retrieve the name of the layers in the current layout `Map 1` item, and formats in one name by line:

```
array_to_string(
 array_foreach(
  map_get( item_variables( 'Map 1' ), 'map_layers' ), -- retrieve the layers␣
↪list
  layer_property( @element, 'name' ) -- retrieve each layer name
 ),
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
 '\n' -- converts the list to string separated by breaklines
)
```

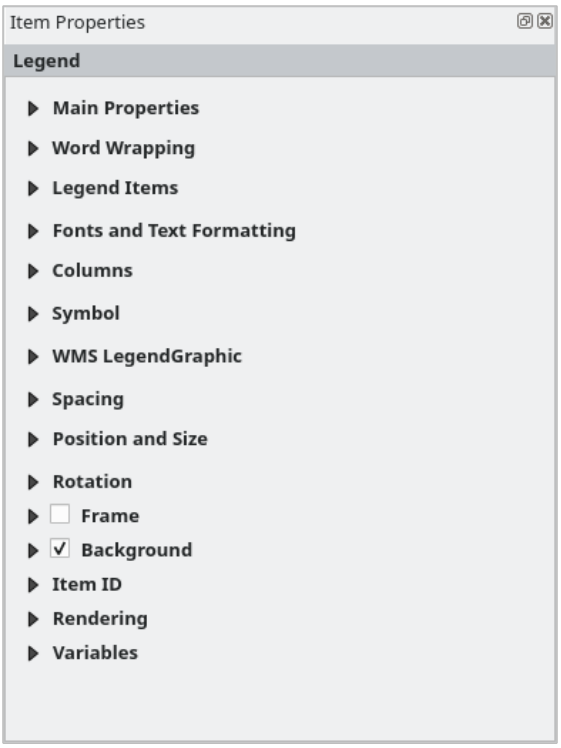
- Display the list of layers with their license strings (usage rights) in a layout `Map 1` item. You need to fill the layers' *Access metadata* properties first.

```
array_to_string( map_credits( 'Map 1', true ) )
```

18.2.5 The Legend Item

The *Legend* item is a box or a table that explains the meanings of the symbols used on the map. A legend is then bound to a map item. You can add a legend item with the *Add Legend* tool following *items creation instructions* and manipulate it the same way as exposed in *Interacting with layout items*.

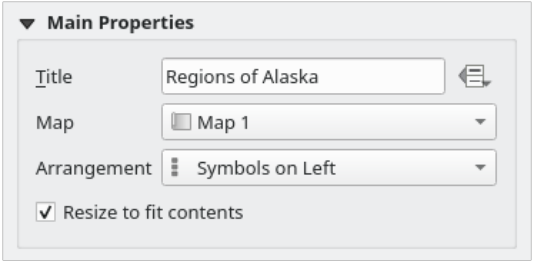
By default, the legend item displays all available layers and can be refined using its *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities (see Fig. 18.27):



18.27 Fig. : Legend Item Properties Panel

Main properties

The *Main properties* group of the legend *Item Properties* panel provides the following functionalities (see Fig. 18.28):



18.28 Fig. : Legend Main properties group

In Main properties you can:

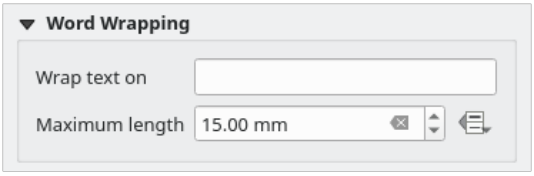
- Change the *Title* of the legend. It can be made dynamic using the *data-defined override* setting, useful for example when generating an atlas;
- Choose which *Map* item the current legend will refer to. By default, the map over which the legend item is drawn is picked. If none, then it falls back to the *reference map*.

> **Pastaba**
>
> *Variables* of the linked map item (@map_id, @map_scale, @map_extent…) are also accessible from data-defined properties of the legend.

- Set the symbols and text placement in the legend: the *Arrangement* can be *Symbols on left* or *Symbols on right*. The default value depends on the locale in use (right-to-left based or not).
- Use *Resize to fit contents* to control whether or not a legend should be automatically resized to fit its contents. If unchecked, then the legend will never resize and instead just stick to whatever size the user has set. Any content which doesn't fit the size is cropped out.

Word wrapping

The *Word wrapping* group of the legend *Item Properties* panel provides the following functionalities:



18.29 Fig. : Legend's word wrapping group

- Wrap the text of the legend on a given character: each time the character appears, it's replaced with a line break;
- Allow legend text to be automatically wrapped after a set line length (in millimeters), preventing very wide auto-generated legends. The *Maximum length* can be set as a static or data-defined value, convenient for dynamic layouts which adjust legend size based on for example page orientation or displayed features, or atlases where you may want to tweak the legend appearance on different pages.

  Examples:

- Control legend width based on layout page orientation

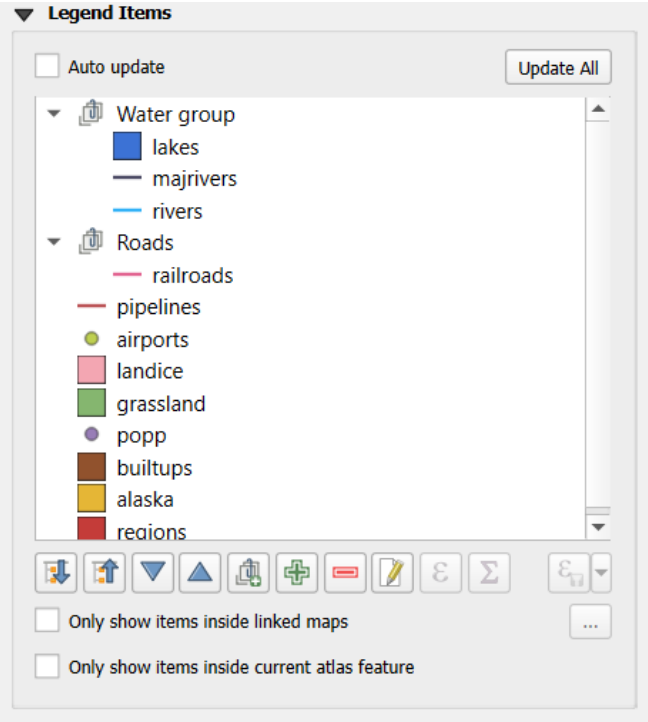
```
IF( @layout_pageheight >= @layout_pagewidth, 35, 80 )
```

- Set the legend column width when a specific layer is displayed in the linked map

```
CASE WHEN
  array_contains(
    map_get( item_variables('legend_1'), 'map_layer_ids' ),
    'a_specific_layer_id' )
THEN 60
END
```

Legend items

The *Legend items* group of the legend *Item Properties* panel provides the following functionalities (see Fig. 18.30):



18.30 Fig. : Legend Items group

- The legend will be updated automatically if *Auto update* is checked. When *Auto update* is unchecked this will give you more control over the legend items. All the icons below the legend items list will be activated.
- The legend items window lists all legend items and allows you to change item order, group layers, remove and restore items in the list, edit layer names and symbology and add a filter.
  - Use the Expand all and Collapse all buttons to respectively expand or collapse all the groups and subgroups in the legend tree. Ensure that the *Auto update* is unchecked to use these buttons.
  - The item order can be changed using the and buttons or with ‚drag-and-drop' functionality. The order can not be changed for WMS legend graphics.
  - Use the button to add a legend group.
  - Use the button to add layers and button to remove groups, layers or symbol classes.

  - The button is used to edit the layer, group name or title. First you need to select the legend item. Double-clicking the item also opens the text box to rename it.
  - The button uses expressions to customize each symbol label of the selected layer (see *Data-define the legend labels*)
  - The button adds a feature count for each class of vector layer.
  - The Filter legend by expression helps you filter which of the legend items of a layer will be displayed, i.e. using a layer that has different legend items (e.g., from a rule-based or categorized symbology), you can specify a boolean expression to remove from the legend tree, styles that have no feature satisfying a condition. Note that the features are nevertheless kept and shown in the layout map item.

  While the default behavior of the legend item is to mimic the *Layers* panel tree, displaying the same groups, layers and classes of symbology, right-click any item offers you options to hide layer's name or raise it as a group or subgroup. In case you have made some changes to a layer, you can revert them by choosing *Reset to defaults* from the contextual menu of the legend entry.

  After changing the symbology in the QGIS main window, you can click on *Update All* to adapt the changes in the legend element of the print layout.

- With the *Only show items inside linked maps*, only the legend items visible in the linked map will be listed in the legend. If you have more than one map you can click on … and select other maps from your layout. This tool remains available when *Auto-update* is active.
- While generating an atlas with polygon features, you can filter out legend items that lie outside the current atlas feature. To do that, check the *Only show items inside current atlas feature* option.

Data-define the legend labels

allows you to add *expressions* to each symbol label of a given layer. New variables (`@symbol_label`, `@symbol_id` and `@symbol_count`) help you interact with the legend entry.

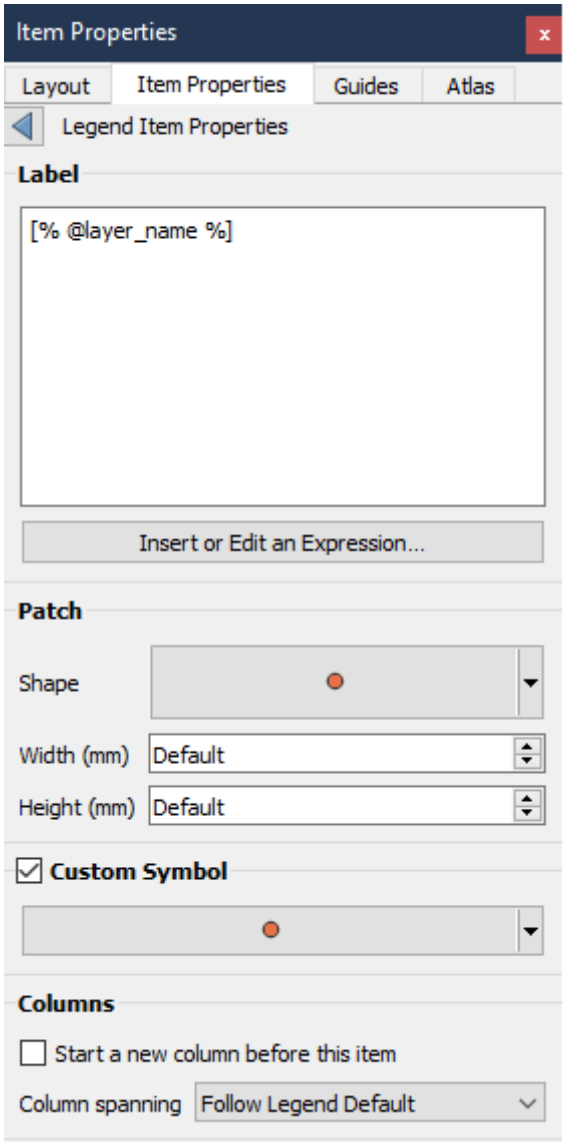
For example, given a `regions` layer categorized by its `type` field, you can append to each class in the legend their number of features and total area, e.g. `Borough (3) - 850ha`:

1. Select the layer entry in the legend tree
2. Press the button, opening the *Expression String Builder* dialog
3. Enter the following expression (*assuming symbol labels have not been edited*):

```
format( '%1 (%2) - %3ha',
        @symbol_label,
        @symbol_count,
        round( aggregate(@layer, 'sum', $area, filter:= "type"=@symbol_label)/
↪10000 )
      )
```

4. Spauskite *Gerai*

Customizing legend items



Legend items can also be customized individually in the *Legend Items Properties*. But these customization can only be done with *Auto update* disabled.

Double-clicking on an item or pressing Edit selected item properties allows for futher customization.

Etiketė

For all item types it allows to modify the label text by typing in or by inserting expressions using the *Insert or Edit an Expression*. Expressions can also be added directly anywhere in the item's label by using the [% expression %] notation.

Columns

The Legend Item Property also allows you to control the column splitting behaviour by forcing the column split to occur after a specific item or all symbols of a layer. Automatic splitting of a layer and its child can also be allowed or blocked on a layer-basis in this widget.

Patch

For items with a symbol, the Legend Item Property allows you to specify the maximum height and width that a symbol can occupy.

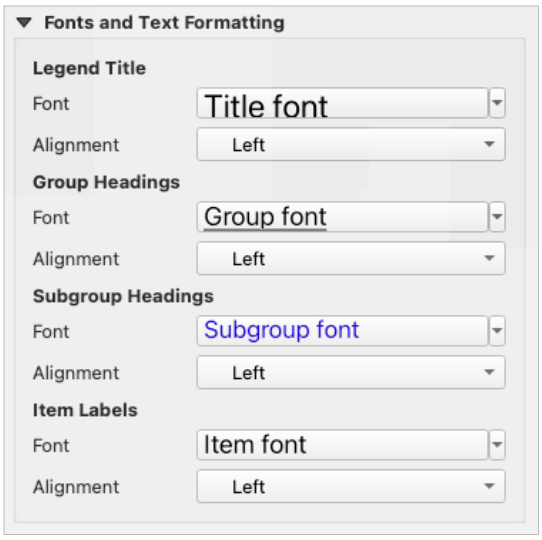
For vector symbols, you can specify a custom shape for the symbol. The shapes are usually defined by an expression to represent the geometry in a simple plane, but those symbols can also be saved in the style manager and imported later. The default symbol for each geometry type can also be controlled via the style manager.

Custom Symbol

A custom symbol can also be specified for vector symbols. This can be useful to tweak the render of a specific symbol, to enhance it in the legend or have a symbol independent from its true symbol preview. This custom symbol will override the legend symbol, but will take into account the symbol *Patch* specified.

Fonts and text formatting

The *Fonts and text formatting* group of the legend *Item Properties* panel provides the following functionalities:



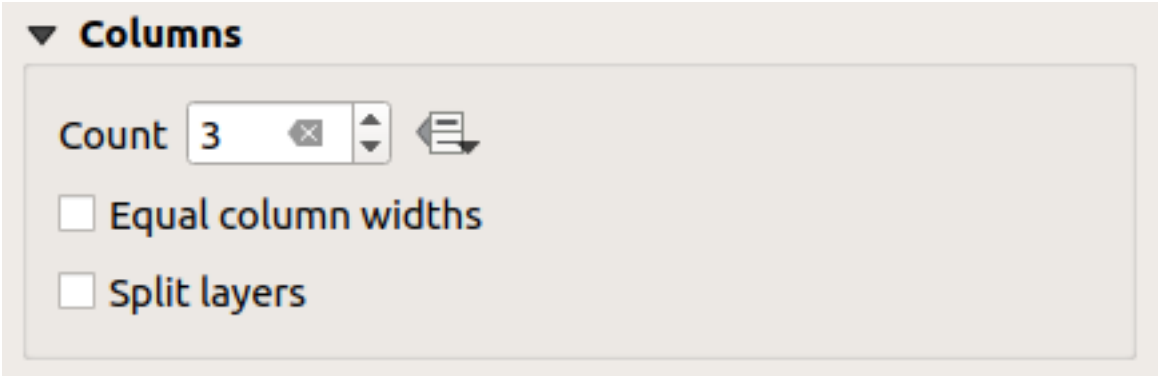
18.31 Fig. : Legend Fonts properties

- You can change the font of the legend title, group, subgroup and item (feature) in the legend item using the *font selector* widget, providing all the capabilities of *text formatting* (font spacing control, mixed HTML formatting, coloring, blending, background, text buffer, shadow, …)
- For each of these levels you can set the text *Alignment*: it can be *Left* (default for left-to-right based locales), *Center* or *Right* (default for right-to-left based locales).

Columns

Under the *Columns* group of the legend *Item Properties* panel, legend items can be arranged over several columns:

- Set the number of columns in the *Count* 1,00 field. This value can be made dynamic e.g., following atlas features, legend contents, the frame size…
- *Equal column widths* sets how legend columns should be adjusted.
- The *Split layers* option allows a categorized or a graduated layer legend to be divided between columns.

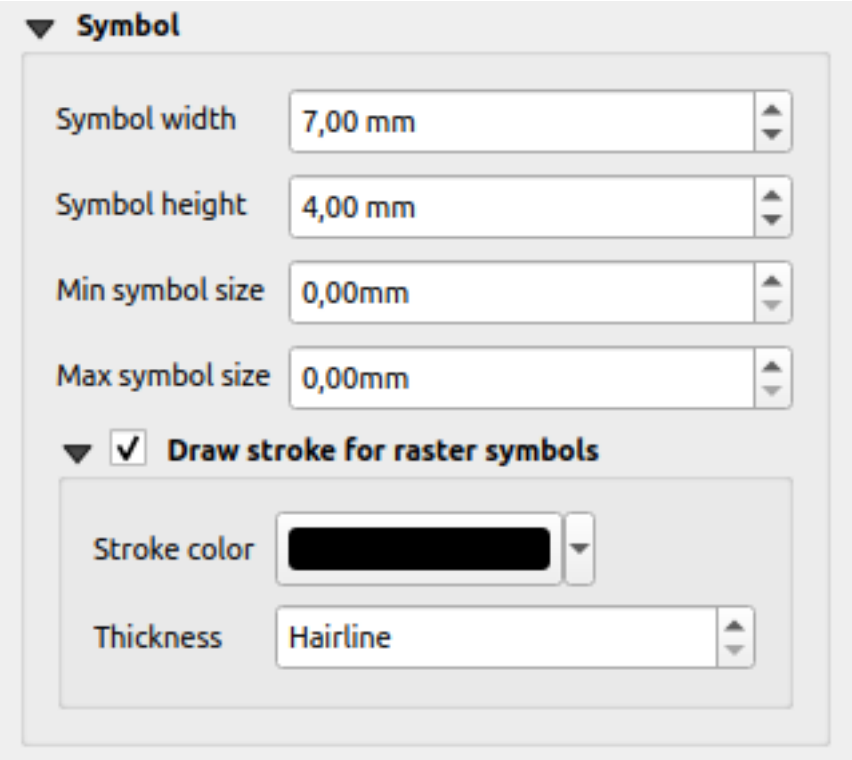


18.32 Fig. : Legend Columns settings

Symbol

The *Symbol* group of the legend *Item Properties* panel configures the size of symbols displayed next to the legend labels. You can:

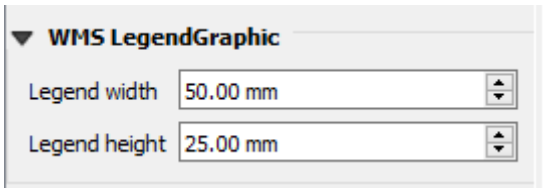
- Set the *Symbol width* and *Symbol height*
- Set the markers' *Min symbol size* and *Max symbol size*: `0.00mm` means there is no value set.
- *Draw stroke for raster symbols*: this adds an outline to the symbol representing the band color of the raster layer; you can set both the *Stroke color* and *Tickness*.



18.33 Fig. : Legend Symbol configuration

WMS LegendGraphic

The *WMS LegendGraphic* section of the legend *Item Properties* panel provide the following functionalities (see Fig. 18.34):

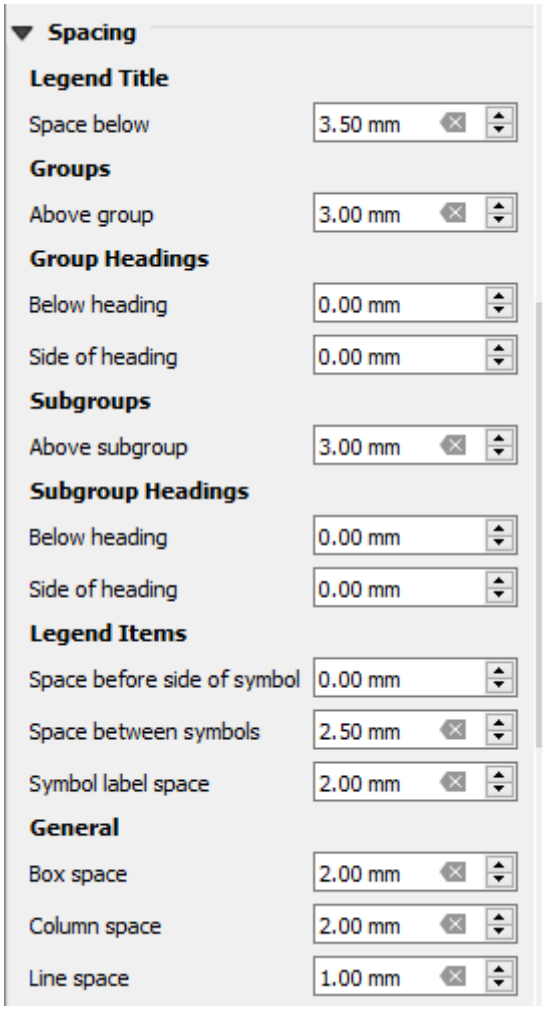


18.34 Fig. : WMS LegendGraphic

When you have added a WMS layer and you insert a legend item, a request will be sent to the WMS server to provide a WMS legend. This Legend will only be shown if the WMS server provides the GetLegendGraphic capability. The WMS legend content will be provided as a raster image.

WMS LegendGraphic is used to be able to adjust the *Legend width* and the *Legend height* of the WMS legend raster image.

Spacing



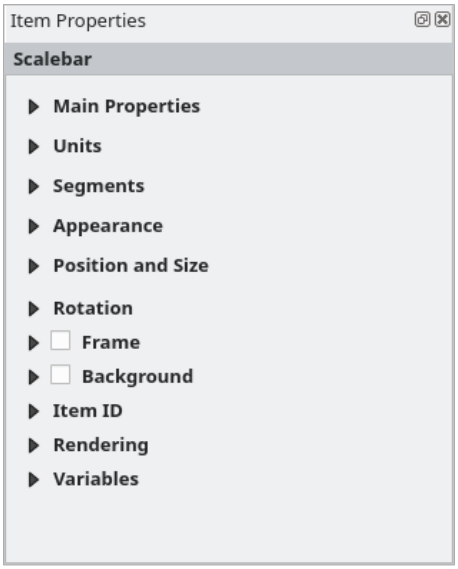
The *Spacing* section allows you to customize the spacing within the legend. Spacing can greatly help denote the groupement of items in the legend and their relation.

Spacing around and before title, groups, subgroups, symbols, labels, boxes, columns and lines can be customized through this dialog.

18.2.6 The Scale Bar Item

Scale bars provide a visual indication of the size of features, and distance between features, on the map item. A scale bar item requires a map item. Use the *Add Scale Bar* tool following *items creation instructions* to add a new scale bar item that you can later manipulate the same way as exposed in *Interacting with layout items*.

By default, a new scale bar item shows the scale of the map item over which it is drawn. If there is no map item below, the *reference map* is used. You can customize it in the *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities (see Fig. 18.35):



18.35 Fig. : Scale Bar Item Properties Panel

Main properties

The *Main properties* group of the scale bar *Item Properties* panel provides the following functionalities (see Fig. 18.36):

18.36 Fig. : Scale Bar Main properties group

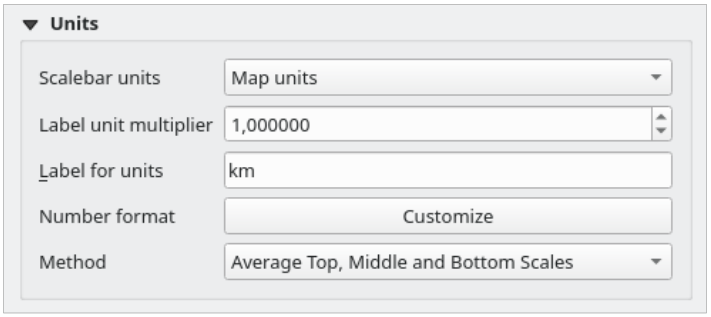
1. First, choose the map the scale bar will be attached to
2. Then, choose the style of the scale bar. Available styles are:
   - **Single box** and **Double box** styles, which contain one or two lines of boxes alternating colors;
   - **Middle**, **Up** or **Down** line ticks;

- **Stepped line** style that draws a stepped line representation of a scalebar
- **Hollow** style that draws a single box with alternating color for the segments, with horizontal lines through alternating segments
- **Numeric**, where the scale ratio is printed (e.g., `1:50000`).

3. Set properties as appropriate

Units

The *Units* group of the scale bar *Item Properties* panel provides the functionalities to set the units of display and some text formatting (see Fig. 18.37):

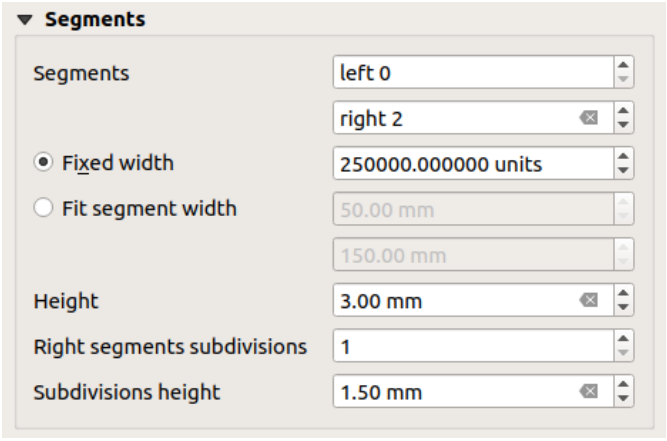


18.37 Fig. : Scale Bar Units group

- Select the units you want to use with *Scalebar units*. There are many possible choices: **Map Units** (the default one), **Meters**, **Feet**, **Miles** or **Nautical Miles**… and some derivatives. Units conversion is handled automatically.
- The *Label unit multiplier* specifies how many scale bar units per labeled unit. Eg, if your scale bar units are set to „meters“, a multiplier of 1000 will result in the scale bar labels in „kilometers“.
- The *Label for units* field defines the text used to describe the units of the scale bar, eg `m` or `km`. This should be matched to reflect the multiplier above.
- Press *Customize* next to *Number format* to have control over all the formatting properties for the numbers in the scale bar, including thousand separators, decimal places, scientific notation, etc. (see *Skaičių formatavimas* for more details). Very useful in the case of making maps for audiences outside of the current QGIS locale, or when you would like to vary the style from the locale defaults (e.g. adding thousands separators when the locale default is to hide them).
- Select the *Method* for scale bar calculation: Depending on the map CRS and extent, the distance reported in the scale bar may not be the same measured over the map. This option, which defaults to the *project's scale calculation method*, helps you specify the one you would like to use for this particular scale bar.

Segments

The *Segments* group of the scale bar *Item Properties* panel provides the functionalities to configure the number and size of segments and subdivisions (see Fig. 18.38):

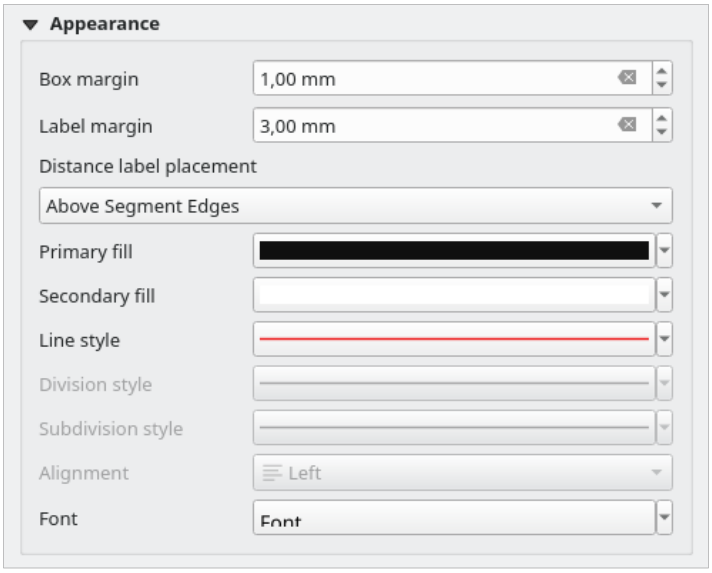


18.38 Fig. : Scale Bar Segments group

- You can define the number of *Segments* that will be drawn at the left and right sides of the `0` of the scale bar:
  - number of subdivisions of a unique segment on the *Left* side
  - number of segments on the *Right* side
- You can set the width of a segment or a range for the total length of the scale bar:
  - Set how long a segment will be in your scale bar units (*Fixed width*)
  - Or limit the total scale bar size in `mm` with *Fit segment width* option. In the latter case, each time the map scale changes, the scale bar is resized (and its label updated) to fit between the upper and lower range set.
- *Height* is used to define the height of the bar.
- *Right segment subdivisions* is used to define the number of sections the right-side segments of the scale bar can have (for *Line Ticks Down*, *Line Ticks Middle* and *Line Ticks Up* scale bar styles) .
- *Subdivision height* is used to define the height of the subdivision segment.

Appearance

The *Appearance* group of the scale bar *Item Properties* panel provides the following functionalities:



18.39 Fig. : Scale Bar Appearance group

You can define how the scale bar will be displayed in its frame.

- *Box margin* : space between text and frame borders
- *Label margin* : space between text and scale bar drawing
- *Distance label placement* defines the position of the text relative to the scale bar segments: *Above Segment Edges*, *Above Segment Centers*, *Below Segment Edges* or *Below Segment Centers*
- *Primary fill* and *Secondary fill* of the scale bar drawing using *fill symbols properties* (color, opacity, patterns, effects…) — for *Single Box*, *Double Box* and *Hollow* styles
- *Line style* of the scale bar drawing using *line symbols properties* (color, stroke, join, cap style, patterns, effects…) — for all but *Numeric* style
- *Division style* and *Subdivision style* respectively for division and subdivision segments in *Line Ticks Up*, *Line Ticks Middle* and *Line Ticks Down* scale bar styles using *line symbols properties* (color, stroke, join, cap style, patterns, effects…)
- *Alignment* puts text on the left, center or right side of the frame (only for *Numeric* scale bar style)
- *Font* to set the *properties* (size, font, color, letter spacing, shadow, background…) of the scale bar label.

Since most of the display properties of the scale bar rely on symbols whose properties can be data-defined, it's possible to render data-defined scale bars.

Example: The following code applied to the bold property of the scale labels will display numbers in bold when they are a multiple of 500:

```
-- returns True (or 1) if the value displayed on the bar
-- is a multiple of 500

@scale_value % 500 = 0
```

18.2.7 Lentelių elementai

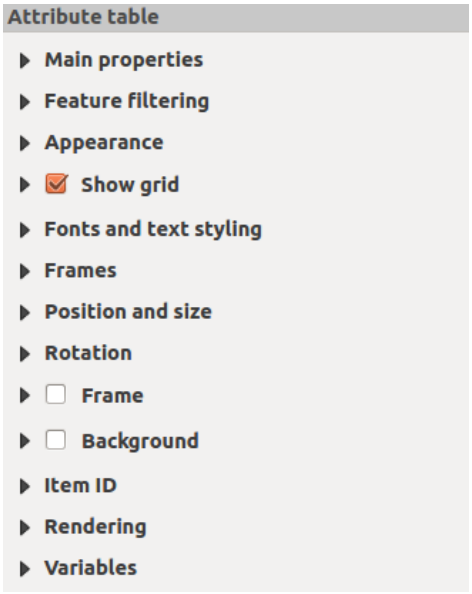
You can use table items to decorate and explain your map:

- *Attribute table*: automatically extracts a subset of the attributes of a layer, based on predefined rules
- *Fixed table*: allows for creation of tables with contents manually entered (i.e. spreadsheet style), so that you can create completely custom tables.

The attribute table item

Any layer in the project can have its attributes shown in the print layout. Use the *Add Attribute Table* tool following *items creation instructions* to add a new table item that you can later manipulate the same way as exposed in *Interacting with layout items*.

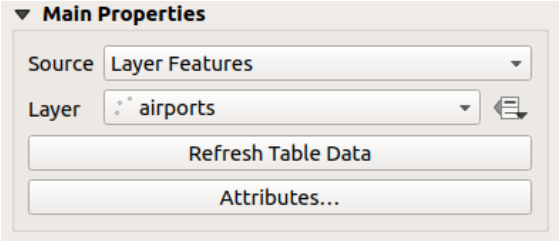
By default, a new attribute table item loads first rows of the first (alphabetically sorted) layer, with all the fields. You can however customize the table thanks to its *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities (see Fig. 18.40):



18.40 Fig. : Attribute table Item Properties Panel

Main properties

The *Main properties* group of the attribute table provides the following functionalities (see Fig. 18.41):



18.41 Fig. : Attribute table Main properties Group

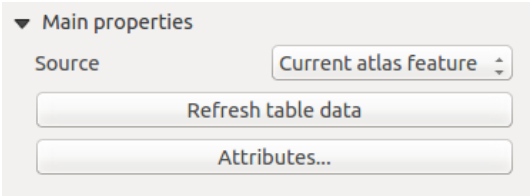
- For *Source* you can by default only select **Layer features** allowing you to select a *Layer* from the vector layers loaded in the project.

  The Data-defined override button near the layer list allows you to dynamically change the layer which is used to populate the table, e.g. you could fill the attribute table with different layer attributes per atlas page. Note that the table structure used (Fig. 18.44) is the one of the layer shown in the *Layer* drop-down list and it is left intact, meaning that setting a data defined table to a layer with different field(s) will result in empty column(s) in the table.

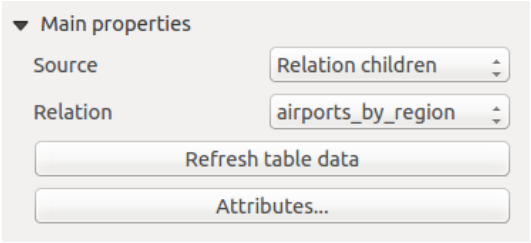
  In case you activate the *Generate an atlas* option in the *Atlas* panel (see *Generate an Atlas*), there are two additional *Source* possible:

  - **Current atlas feature** (see Fig. 18.42): you won't see any option to choose the layer, and the table item will only show a row with the attributes from the current feature of the atlas coverage layer.
  - and **Relation children** (see Fig. 18.43): an option with the relation names will show up. This feature can only be used if you have defined a *relation* using your atlas coverage layer as parent, and the table will show the children rows of the atlas coverage layer's current feature.

- The button *Refresh Table Data* can be used to refresh the table when the actual contents of the table has changed.

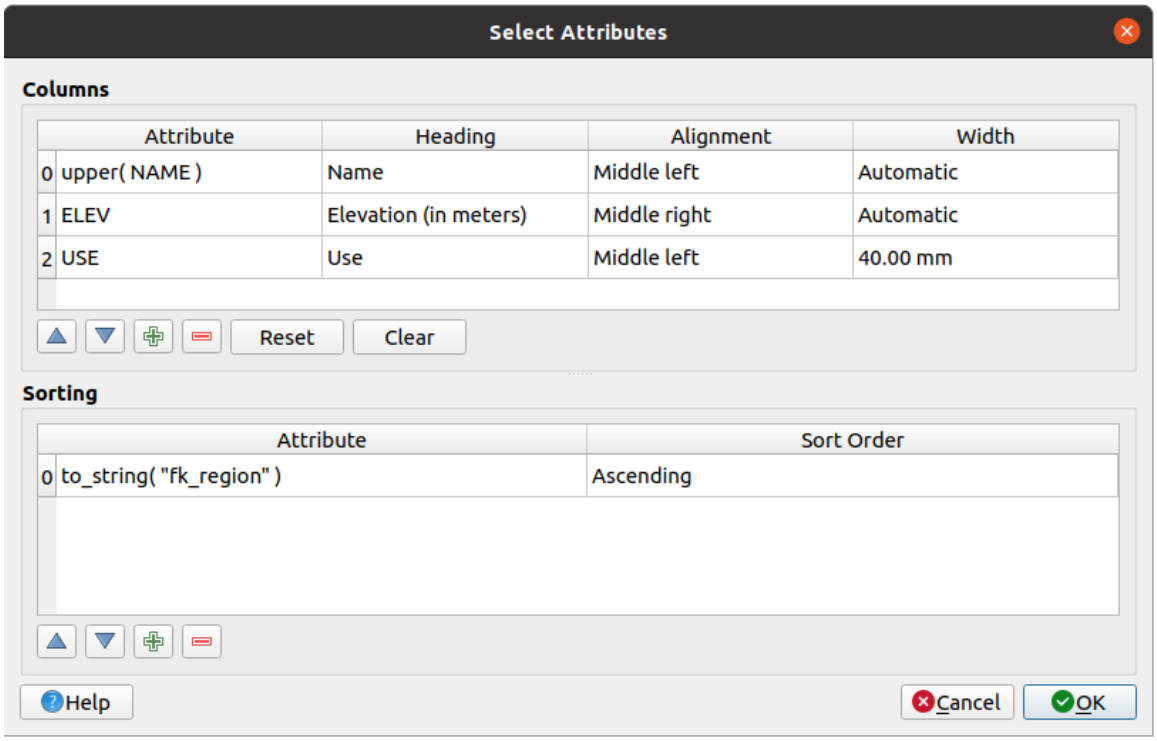


18.42 Fig. : Attribute table Main properties for ‚Current atlas feature‘



18.43 Fig. : Attribute table Main properties for ‚Relation children‘

- The button *Attributes…* starts the *Select Attributes* dialog, (see Fig. 18.44) that can be used to change the visible contents of the table. The upper part of the window shows the list of the attributes to display and the lower part helps you sort the data.



18.44 Fig. : Attribute table Select attributes Dialog

In the *Columns* section you can:

- Move attributes up or down the list by selecting the rows and then using the and buttons to shift the rows. Multiple rows can be selected and moved at any one time.
- Add an attribute with the button. This will add an empty row at the bottom of the table where you can select a field to be the attribute value or create an attribute via a regular expression.

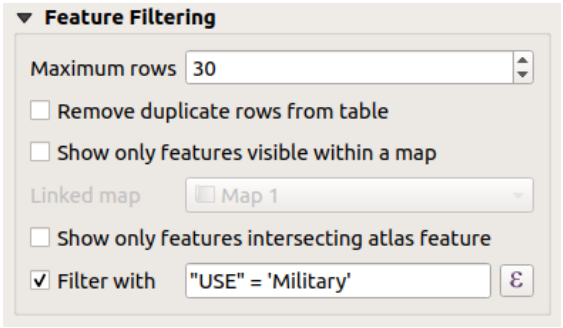
- Remove an attribute with the button. Multiple rows can be selected and removed at any one time.
- Reset the attribute table back to its default state with the *Reset* button.
- Clear the table using the *Clear* button. This is useful when you have a large table but only want to show a small number of attributes. Instead of manually removing each row, it may be quicker to clear the table and add the rows needed.
- Cell headings can be altered by adding the custom text in the *Heading* column.
- Cell alignment can be managed with the *Alignment* column which will dictate the texts position within the table cell.
- Cell width can be manually managed by adding custom values to the *width* column.

In the *Sorting* section you can:

- Add an attribute to sort the table with: press the button and a new empty row is added. Insert a field or an expression in the *Attribute* column and set the *Sort order* to **Ascending** or **Descending**.
- Select a row in the list and use the and buttons to change the sort priority on attribute level. Selecting a cell in the *Sort Order* column helps you change the sorting order of the attribute field.
- Use the button to remove an attribute from the sorting list.

Geoobjektų filtravimas

The *Feature filtering* group of the attribute table provides the following functionalities (see Fig. 18.45):



18.45 Fig. : Attribute table Feature filtering Group

Jūs galite:

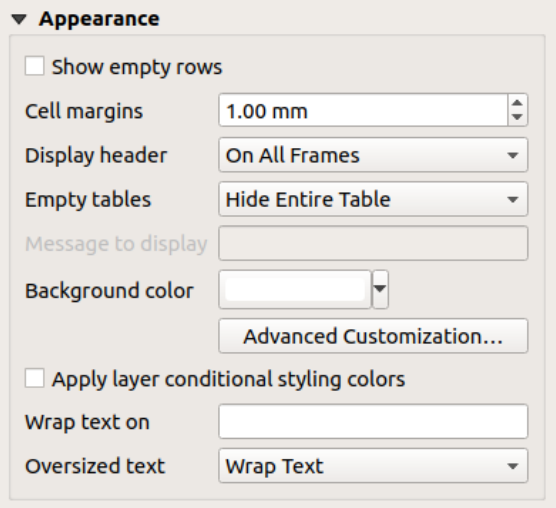
- Define the *Maximum rows* to be displayed.
- Activate *Remove duplicate rows from table* to show unique records only.
- Activate *Show only visible features within a map* and select the corresponding *Linked map* whose visible features attributes will be displayed.
- Activate *Show only features intersecting Atlas feature* is only available when *Generate an atlas* is activated. When activated it will show a table with only the features which intersect the current atlas feature.
- Activate *Filter with* and provide a filter by typing in the input line or insert a regular expression using the given expression button. A few examples of filtering statements you can use when you have loaded the airports layer from the Sample dataset:
  - `ELEV > 500`

- `NAME = 'ANIAK'`
- `NAME NOT LIKE 'AN%'`
- `regexp_match( attribute( $currentfeature, 'USE' ) , '[i]')`

The last regular expression will include only the airports that have a letter ‚i' in the attribute field ‚USE'.

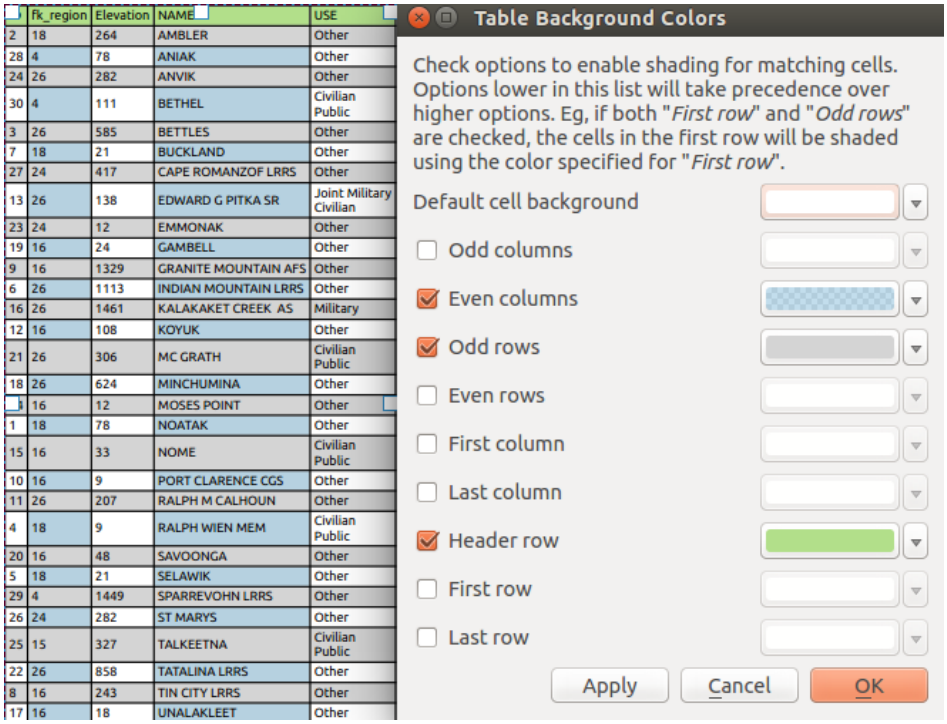
Appearance

The *Appearance* group of the attribute table provides the following functionalities (see Fig. 18.46):



18.46 Fig. : Attribute table appearance Group

- Click *Show empty rows* to fill the attribute table with empty cells. This option can also be used to provide additional empty cells when you have a result to show!
- With *Cell margins* you can define the margin around text in each cell of the table.
- With *Display header* you can select from a list one of ‚On first frame', ‚On all frames' default option, or ‚No header'.
- The option *Empty table* controls what will be displayed when the result selection is empty.
  - **Draw headers only**, will only draw the header except if you have chosen ‚No header' for *Display header*.
  - **Hide entire table**, will only draw the background of the table. You can activate *Don't draw background if frame is empty* in *Frames* to completely hide the table.
  - **Show set message**, will draw the header and adds a cell spanning all columns and display a message like ‚No result' that can be provided in the option *Message to display*
- The option *Message to display* is only activated when you have selected **Show set message** for *Empty table*. The message provided will be shown in the table in the first row, when the result is an empty table.
- With *Background color* you can set the background color of the table using the *color selector* widget. The *Advanced customization* option helps you define different background colors for each cell (see Fig. 18.47)



18.47 Fig. : Attribute table Advanced Background Dialog

- *Apply layer conditional styling colors*: the *conditional table formatting* present in the layer is applied inside the layout attribute table (background color, font family and properties such as bold, italic, strikeout, underline, color, …). Conditional formatting rules take precedence over other layout table formatting settings, e.g. they will override other cell background color settings such as alternating row colors.
- With the *Wrap text on* option, you can define a character on which the cell content will be wraped each time it is met
- With *Oversized text* you define the behavior when the width set for a column is smaller than its content's length. It can be **Wrap text** or **Truncate text**.

Pastaba

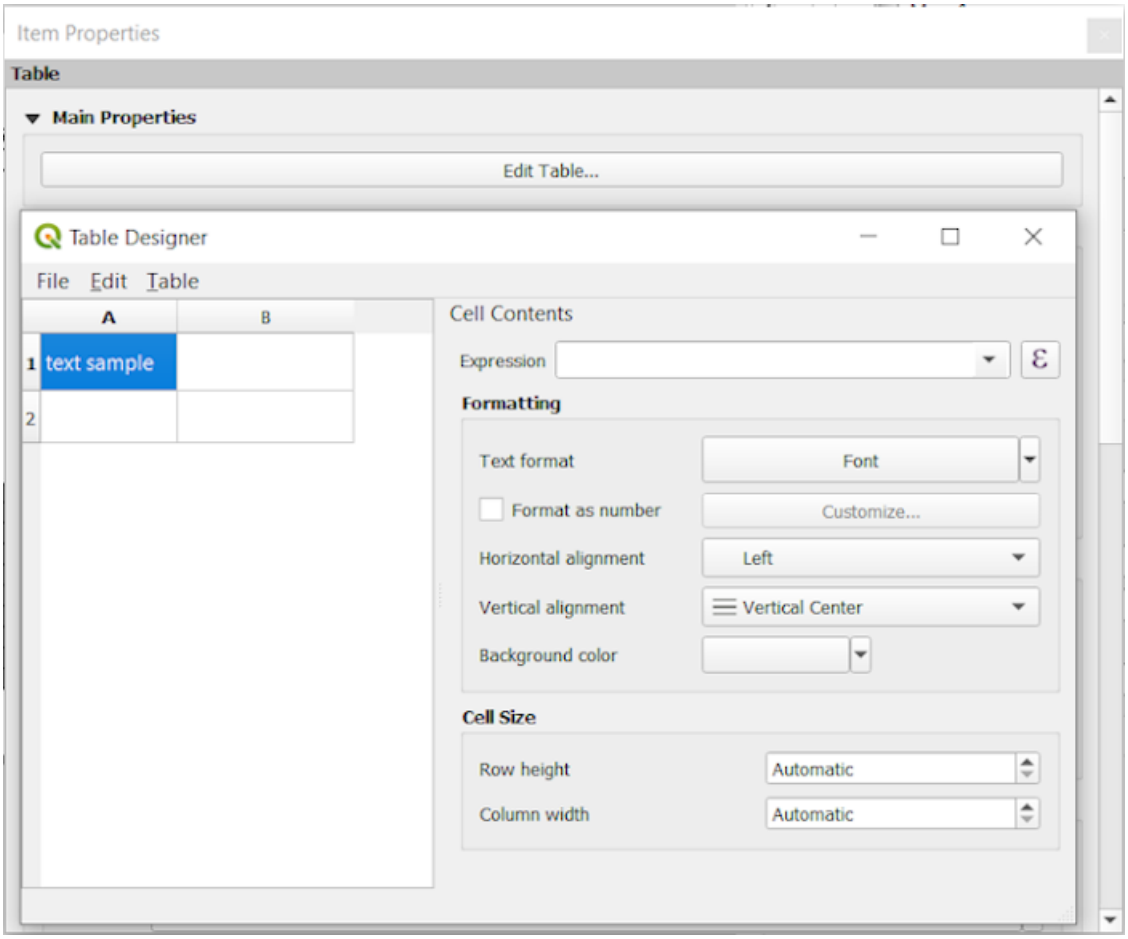
More properties of the attribute table item are described in the *Tables common functionalities* section.

The fixed table item

Additional information about the map can be inserted manually into a table by choosing *Add Fixed Table* and by following *items creation instructions* to add a new table item that you can later manipulate the same way as exposed in *Interacting with layout items*.

By default, an empty table with two minimized columns and rows appears in the map layout. You have to customize the table in the *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities:

Main properties



18.48 Fig. : Fixed table Item Properties Panel with Table designer

In *Main properties*, press *Edit table…* to open the *Table designer* dialog where you can build your table (double-clicking the table item will also open the editor dialog):

- On the left hand of the dialog, a table frame displays the active table item with its content and settings. Click or double-click on a cell and write a text to modify its contents. Press `Ctrl+Enter` to add linebreak (for a multiline text). The logic for selecting multiple cells in the table is the same as in the vector layer *attribute table*. Grab and move a row or column separator to resize the row above or the column before. A double-click on the separator will resize the row above or the column before to fit their contents. A right-click over column or row headers shows a contextual menu to insert columns or rows around the selection or to delete them.
- Through the menus on top of the dialog, it is possible to:
  - *File* ► *Import Content From Clipboard*: it overrides contents of the table.
  - *File* ► *Close editor*: it closes the *Table Designer* dialog.
  - From the *Edit* menu, you can work with selection functionalities for rows and columns:
    - *Select all* cells in the table
    - *Select columns* or *Select Rows*: when cells are selected in the table, you can extend the selection respectively to their column(s) or row(s)
    - *Clear Cells*: deletes the content of the selected cells
  - The *Table* menu is the place where you design the structure of the table. You can:

    * *Insert rows* ► above or below the selection
    * *Insert columns* ► before or after the selection
    * *Delete Rows* or *Delete Columns* of selected cells
    * *Merge selected cells*: select multiple cells in a rectangular shape and you can merge them and concatenate their contents into the top left cell (whose styling is also applied to the merged cell, unless there is a last row/column cell in the selection).
    * *Split selected cells*: select a merged cell and you can split it back to its individual cells. The current text is kept in the top left cell, and the other cells are filled with their contents before they get merged. They are also reapplied their original styling.
    * *Include Header Row* whose styling can only be controlled from the *Fonts and text styling* widget
- Select the cells and, on the right of the table frame, you can provide the *Cell Contents*:
  - Either manually enter the contents of each cell, or use an *Expression* to automatically populate it.
  - Under the *Formatting* group, set specific options that apply to the selection and take precedence over the *global table content styling*:
    * by choosing the *Text format*
    * by *Format as number* (several formats are available)
    * by defining the *Horizontal alignment* and the *Vertical alignment*
    * by choosing a *Background color*
  - Define the *Cell Size* with *Row height* and *Column width*.

Appearance

The *Appearance* group of the fixed table provides the following functionalities:

- Click *Show empty rows* to fill the attribute table with empty cells.
- With *Cell margins* you can define the margin around text in each cell of the table.
- With *Display header* you can select from a list one of ‚On first frame', ‚On all frames' default option, or ‚No header'.
- With *Background color* you can set the background color of the table using the *color selector* widget. The *Advanced customization* option helps you define different background colors for each cell.
- With *Oversized text* you define the behavior when the width set for a column is smaller than its content's length. It can be **Wrap text** or **Truncate text**.

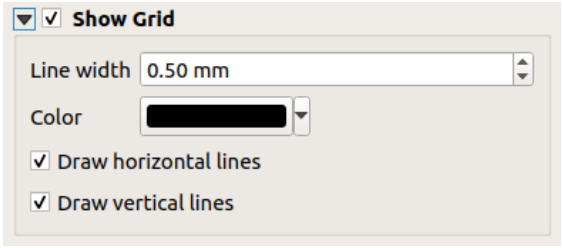
Pastaba

More properties of the fixed table item are described in the *Tables common functionalities* section.

Tables common functionalities

Show grid

The *Show grid* group of the table items provides the following functionalities (see Fig. 18.49):



18.49 Fig. : Attribute table Show grid Group

- Activate *Show grid* when you want to display the grid, the outlines of the table cells. You can also select to either *Draw horizontal lines* or *Draw vertical lines* or both.
- With *Line width* you can set the thickness of the lines used in the grid.
- The *Color* of the grid can be set using the color selection widget.

Fonts and text styling

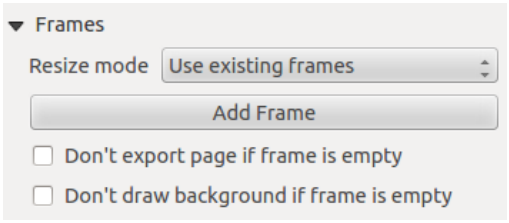
The *Fonts and text styling* group of the table items provides the following functionalities (see Fig. 18.50):

18.50 Fig. : Attribute table Fonts and text styling Group

- You can define *Font* properties for *Table heading* and *Table contents*, using the advanced *text settings* widget (with buffer, shadow, paint effects, transparence, background, coloring, …). Note that these changes do not affect the cells that have custom font assigned, either from the *Appearance* section or the *Table Designer* dialog. Only cells with the default rendering are overwritten.
- For *Table heading* you can additionally set the *Alignment* to `Follow column alignment` or override this setting by choosing `Left`, `Center` or `Right`. The column alignment is set using the *Select Attributes* dialog (see Fig. 18.44).

Frames

The *Frames* group of the table item properties provides the following functionalities (see Fig. 18.51):



18.51 Fig. : Attribute table Frames Group

- With *Resize mode* you can select how to render the attribute table contents:
  - `Use existing frames` displays the result in the first frame and added frames only.
  - `Extend to next page` will create as many frames (and corresponding pages) as necessary to display the full selection of attribute table. Each frame can be moved around on the layout. If you resize a frame, the resulting table will be divided up between the other frames. The last frame will be trimmed to fit the table.
  - `Repeat until finished` will also create as many frames as the *Extend to next page* option, except all frames will have the same size.
- Use the *Add Frame* button to add another frame with the same size as selected frame. The result of the table that will not fit in the first frame will continue in the next frame when you use the Resize mode `Use existing frames`.
- Activate *Don't export page if frame is empty* prevents the page to be exported when the table frame has no contents. This means all other layout items, maps, scalebars, legends etc. will not be visible in the result.
- Activate *Don't draw background if frame is empty* prevents the background to be drawn when the table frame has no contents.

18.2.8 The Marker, Picture and North Arrow Items

- *The Picture Item*
  - *Main properties*
  - *Size and placement*
  - *Image rotation*
- *The North Arrow Item*
- *The Marker Item*

Along with the map or legend items in the print layout, you may want to decorate your realization with images or annotations. QGIS provides different tools to achieve this:

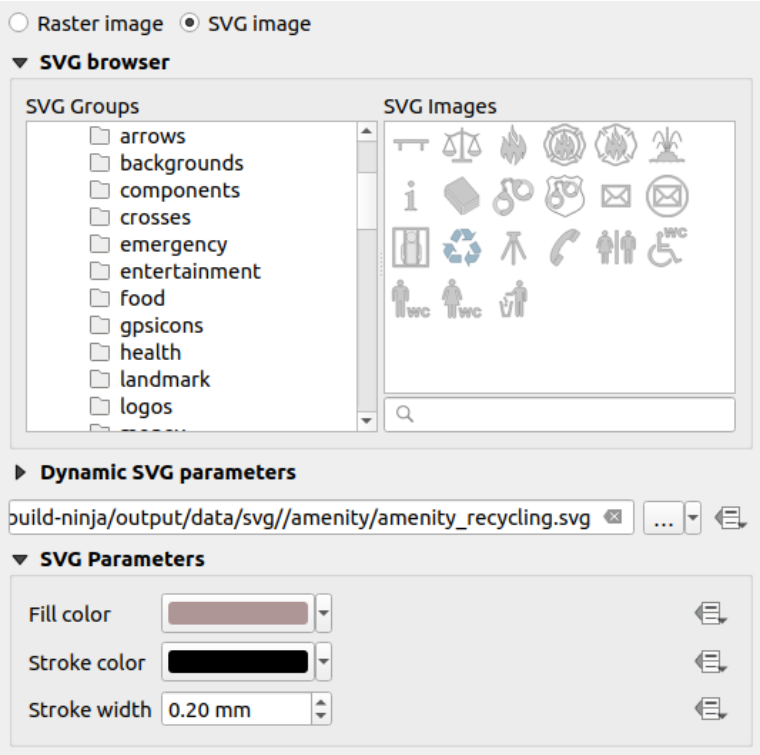
- the *picture item*: decorates the layout with an image raster or SVG file (e.g. logos, pictures, north arrows, …)
- the *north arrow item*: a picture item predefined with a north arrow image
- the *marker item*: decorates the layout with QGIS vector *symbols*. It can be used to place markers over a map item or for creation of advanced custom legends.

The Picture Item

You can add a picture by dragging it from your file manager onto the canvas, pasting it directly into the layout by using `Ctrl+V` or *Edit* ► *Paste* and by using the Add Picture, following *items creation instructions*. Then you can manipulate it, as explained in *Interacting with layout items*.

When using Add Picture, the picture item will be a blank frame that you can customize using its *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities:

Main properties



18.52 Fig. : Picture Item Properties panel

The picture item supports two types of images:

- *Raster Image*: a file selector widget can be used to fetch the data. Use the … Browse button to select a file on your computer or enter the path directly in the text field. You can even provide a remote URL that points to a picture. The associated image can also be *embedded* in the layout.

  Use the data defined override button to set the image source from a feature attribute or using a regular expression.

- *SVG Image*: using by default the SVG libraries provided in *Settings* ► *Options* ► *System* ► *SVG Paths*. You can however use any other file, and the file selection follows the same rules as for the raster image. The SVG parameters can as well be set dynamic.

  The QGIS provided (default) `.SVG` files are customizable, meaning that you can easily apply other *Fill color*, *Stroke color* (including opacity) and *Stroke width* than the original, using their corresponding feature in the *SVG Parameters* group. These properties can also be *data-defined*.

  If you add an `.SVG` file that does not enable these properties, you may need to add the following tags to the file in order to add support e.g. for transparency:

  – *fill-opacity="param(fill-opacity)"*

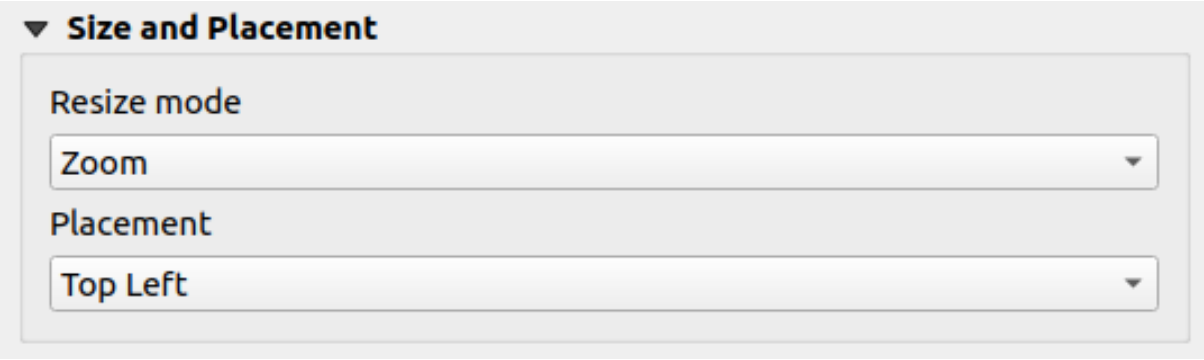
- *stroke-opacity="param(outline-opacity)"*

More details at *Parametrizuojami SVG*.

Pastaba

Drag-and-drop an image file (raster or SVG) into the layout page will create a layout picture item with corresponding settings.

Size and placement



18.53 Fig. : Layout pictures size and placement properties

With the *Resize mode* option, you can set how the image is displayed when the frame is resized:

- `Zoom`: enlarges/reduces the image to the frame while maintaining the aspect ratio of picture
- `Stretch`: stretches the image to fit inside the frame
- `Clip`: use this mode for raster images only, it sets the size of the image to the original image size without scaling, and the frame is used to clip the image. So only the part of the image that is inside the frame will be visible.
- `Zoom and resize frame`: enlarges the image to fit the frame, and then resizes frame to fit the resulting image dimensions
- `Resize frame to image size`: sets the size of the frame to match the original size of the image (no scaling)

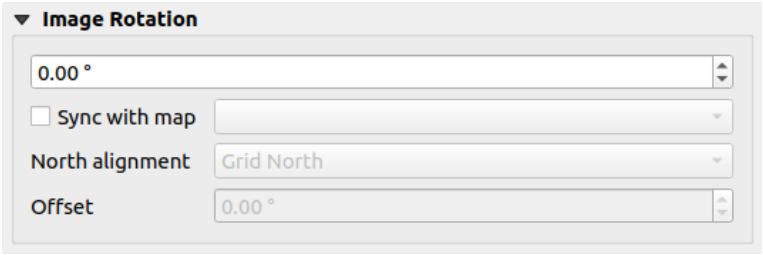
Depending on the selected *Resize mode*, the *Placement* and *Image rotation* options may be disabled. *Placement* lets you select the position of the image inside its frame (top/middle/bottom and left/center/right).

Image rotation

Images can be rotated with the *Image rotation* field. Activating the *Sync with map* checkbox synchronizes the rotation of the image with the rotation applied to the selected map item. This is a convenient feature to make any picture behave as a north arrow. The *North alignment* can be:

- **Grid north**: the direction of a grid line which is parallel to the central meridian of the national/local grid
- **True north**: direction of a meridian of longitude.

You can also apply a declination *Offset* to the picture rotation.



18.54 Fig. : Layout pictures image rotation properties

The North Arrow Item

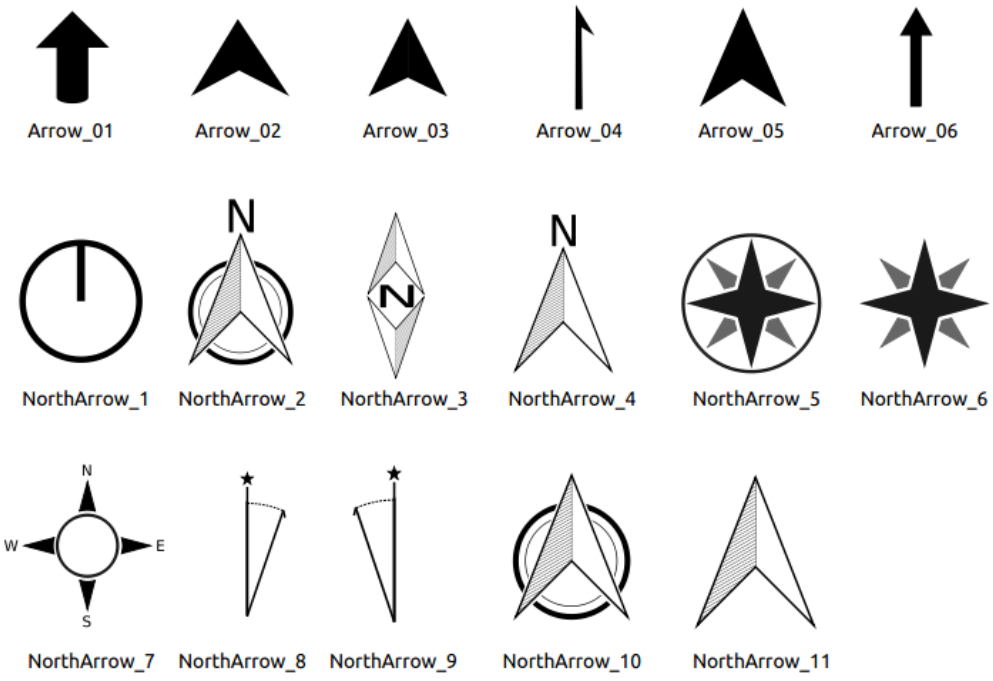
You can add a north arrow with the Add North Arrow button, following *items creation instructions* and manipulate it the same way as exposed in *Interacting with layout items*.

Since north arrows are images, the *North Arrow* item has the same properties as the *picture item*. The main differences are:

- A default north arrow is used when adding the item, instead of a blank frame
- The north arrow item is synced with a map item by default: the *Sync with map* property is the map over which the north arrow item is drawn. If none, it falls back to the *reference map*.

Pastaba

Many of the north arrows do not have an ‚N' added in the north arrow. This is done on purpose, since there are languages that do not use an ‚N' for North.



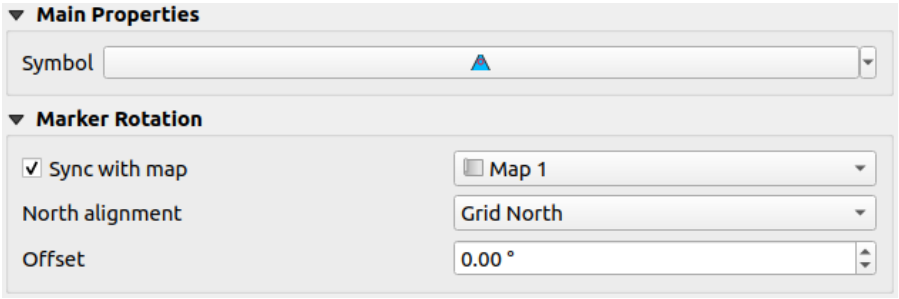
18.55 Fig. : North arrows available for selection in provided SVG library

The Marker Item

To add a marker item, select the Add Marker button, and click on the page. A default point marker symbol is added. Then you can manipulate it, as explained in *Interacting with layout items*. But note that unlike most of the other items, you resize the item given that its size is controlled by the embedded symbols properties.

The marker item can be customized from the *Item Properties* panel. Other than the *items common properties*, you can also:

- modify the *Symbol*, relying on all the symbol *widget capabilities*
- sync the marker item rotation with the map's (see *Image rotation*), acting as a north arrow. The map rotation is added to any existing marker symbol level rotation (so .e.g if you have to rotate the triangle marker 90° to get it pointing straight up, it will still work nicely in north arrow mode!)



18.56 Fig. : The marker item custom properties

18.2.9 The Elevation Profile Item

The Elevation Profile item is used to display an *elevation profile view* in a layout. Use the *Add Elevation Profile* button, and follow *items creation instructions* to add a new elevation profile item that you can later manipulate the same way as demonstrated in *Interacting with layout items*.

A new elevation profile item has default settings that make it render an empty chart. You can customize its properties in the *Item Properties* panel. In addition to the *common properties*, this feature has the following functionalities:

The Elevation profile *Item Properties* panel embeds a top toolbar with the following functionalities:

- Update elevation profile to refresh the item rendering
- Copy from elevation profile: a drop-down menu to select an elevation profile view from. The view settings are applied to the layout elevation profile item, and can be later modified.

Layers

Under the *Layers* group, check in the tree view the layers you would like to render in the profile item. Remember to properly configure the *Elevation* properties of the selected layers.

Profile curve

- *Controlled by atlas*: the *profile curve* will be taken from the current atlas feature and the elevation profile view updated as you walk through the atlas features. This is currently supported for an active layout atlas or report, using a line geometry type coverage layer.
- The *Tolerance* distance, which can be data-defined, helps you control how far a feature of the visible layers should be from the profile curve in order to display in the layout elevation view. Only point features are currently returned.

Show Subsections Indicator

It is possible to highlight the position of the vertices of the curve on the Elevation Profile. Select the *Show Subsections Indicator* to display vertical lines at the distances of the vertices. Use the *Lines* drop-down list to set the symbol, color, opacity and width of the Subsections Indicator lines.

Chart ranges

A layout elevation profile item does not necessarily display the full extent of the elevation profile view it is based on. You can limit the area to render, providing:

- X ašyje *Minimalus atstumas* ir *Maksimalus atstumas* nuo profilio kreivės pradžios taško
- Y ašyje *Minimalus aukštis* ir *Maksimalus aukštis*

Distance and elevation axes

The *Distance axis* and *Elevation axis* groups give options to tweak the grid over the elevation profile item, respectively on the X and Y axes:

- the distance display *Unit*, allowing to override the map canvas unit
- the graduation on the axis with both a *Major interval* and *Minor interval*
- the line symbols to apply to the corresponding *Major grid lines* and *Minor grid lines*
- how regular the graduation items should be labeled (*Label interval*) as well as their *Label format* and *Label font*
- the *Distance labels*: configures whether the distance unit symbol should be placed next to *Every value*, *First value*, *Last value*, *First and last values*, or skipped (*None*)

Chart area

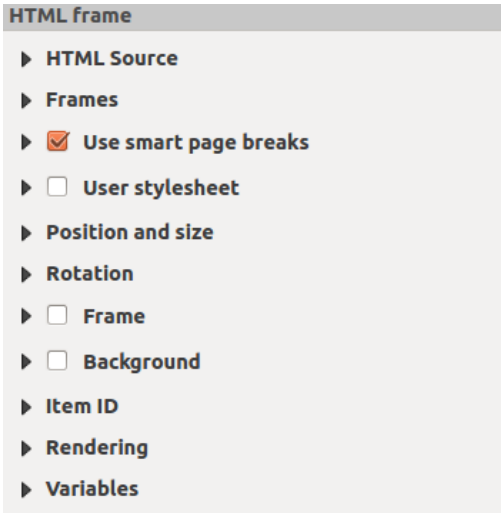
Under *Chart area*, you can configure the rendering of the area in which the elevation profile plot is actually displayed:

- a *Background* fill symbol
- a *Border* line symbol
- the margins from the elevation profile item border

18.2.10 The HTML Frame Item

It is possible to add a frame that displays the contents of a website or even create and style your own HTML page and display it! You can add a picture with the *Add HTML* following *items creation instructions* and manipulate it the same way as exposed in *Interacting with layout items*. Note that the HTML scale is controlled by the layout export resolution at the time the HTML frame is created.

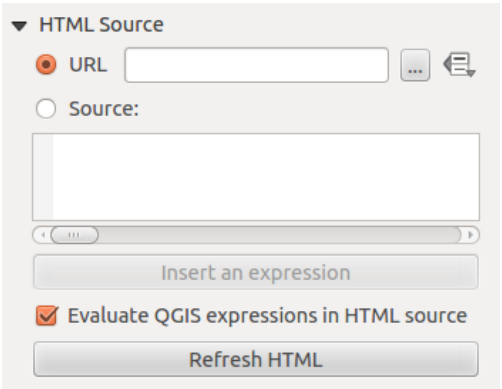
The HTML item can be customized using its *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities (see Fig. 18.57):



18.57 Fig. : HTML Frame, the Item Properties Panel

HTML Source

The *HTML Source* group of the HTML frame *Item Properties* panel provides the following functionalities (see Fig. 18.58):



18.58 Fig. : HTML frame, the HTML Source properties

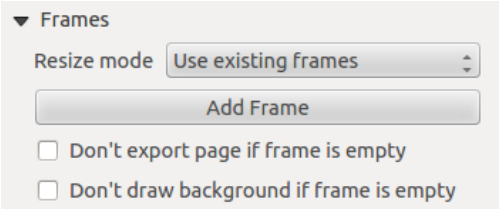
- In *URL* you can enter the URL of a webpage you copied from your Internet browser or select an HTML file using the … Browse button. There is also the option to use the Data-defined override button, to provide a URL from the contents of an attribute field of a table or using a regular expression.
- In *Source* you can enter text in the textbox with some HTML tags or provide a full HTML page.
- The *Insert or Edit an Expression…* button can be used to add an expression like `[%Year($now)%]` in the Source textbox to display the current year. This button is only activated when radiobutton *Source* is selected.

After inserting the expression click somewhere in the textbox before refreshing the HTML frame, otherwise you will lose the expression.

- Activate *Evaluate QGIS expressions in HTML code* to see the result of the expression you have included, otherwise you will see the expression instead.
- Use the *Refresh HTML* button to refresh the HTML frame(s) and see the result of changes.

Frames

The *Frames* group of the HTML frame *Item Properties* panel provides the following functionalities (see Fig. 18.59):

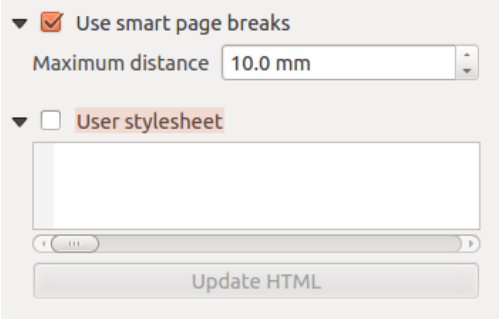


18.59 Fig. : HTML frame, the Frames properties

- With *Resize mode* you can select how to render the HTML contents:
  - `Use existing frames` displays the result in the first frame and added frames only.
  - `Extend to next page` will create as many frames (and corresponding pages) as necessary to render the height of the web page. Each frame can be moved around on the layout. If you resize a frame, the webpage will be divided up between the other frames. The last frame will be trimmed to fit the web page.
  - `Repeat on every page` will repeat the upper left of the web page on every page in frames of the same size.
  - `Repeat until finished` will also create as many frames as the `Extend to next page` option, except all frames will have the same size.
- Use the *Add Frame* button to add another frame with the same size as selected frame. If the HTML page does not fit in the first frame it will continue in the next frame when you use *Resize mode* or *Use existing frames*.
- Activate *Don't export page if frame is empty* prevents the page from being exported when the frame has no HTML contents. This means all other layout items, maps, scale bars, legends etc. will not be visible in the result.
- Activate *Don't draw background if frame is empty* prevents the HTML frame being drawn if the frame is empty.

Use smart page breaks and User style sheet

The *Use smart page breaks* dialog and *User style sheet* dialog of the HTML frame *Item Properties* panel provides the following functionalities (see Fig. 18.60):



18.60 Fig. : HTML frame, Use smart page breaks and User style sheet properties

- Activate *Use smart page breaks* to prevent the html frame contents from breaking mid-way a line of text so it continues nice and smooth in the next frame.
- Set the *Maximum distance* allowed when calculating where to place page breaks in the html. This distance is the maximum amount of empty space allowed at the bottom of a frame after calculating the optimum break location. Setting a larger value will result in better choice of page break location, but more wasted space at the bottom of frames. This is only used when *Use smart page breaks* is activated.
- Activate *User style sheet* to apply HTML styles that often is provided in cascading style sheets. An example of style code is provided below to set the color of `<h1>` header tag to green and set the font and font size of text included in paragraph tags `<p>`.

```
h1 {color: #00ff00;
}
p {font-family: "Times New Roman", Times, serif;
   font-size: 20px;
}
```

- Use the *Update HTML* button to see the result of the style sheet settings.

18.2.11 The Shape Items

QGIS provides a couple of tools to draw regular or more complex shapes over the print layout.

Pastaba

Unlike other print layout items, you can not style the frame nor the background color of the shapes bounding frame (set to transparent by default).

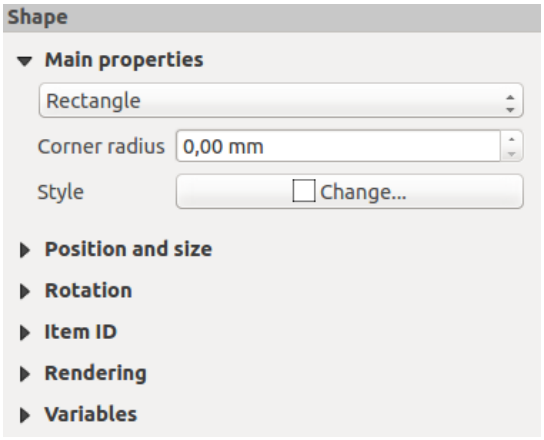
The Regular Shape Item

The *Shape* item is a tool that helps to decorate your map with regular shapes like triangle, rectangle, ellipse… You can add a regular shape using the Add Shape tool which gives access to particular tools like Add Rectangle, Add Ellipse and Add Triangle. Once you have selected the appropriate tool, you can draw the item following *items creation instructions*. Like other layout items, a regular shape can be manipulated the same way as exposed in *Interacting with layout items*.

Pastaba

Holding down the `Shift` key while drawing the basic shape with the click and drag method helps you create a perfect square, circle or triangle.

The default shape item can be customized using its *Item Properties* panel. Other than the *items common properties*, this feature has the following functionalities (see Fig. 18.61):



18.61 Fig. : Shape Item Properties Panel

The *Main properties* group shows and allows you to switch the type of the shape item (**Ellipse**, **Rectangle** or **Triangle**) inside the given frame.

You can set the style of the shape using the advanced *symbol* and *color* selector widget…

For the rectangle shape, you can set in different units the value of the *Corner radius* to round of the corners.

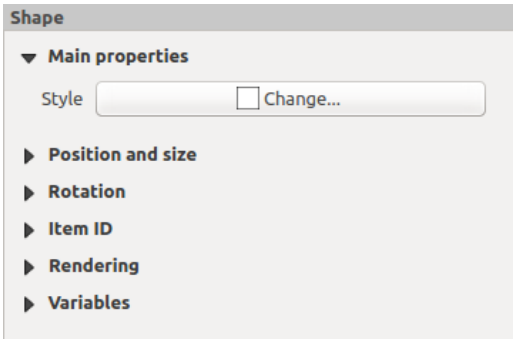
The Node-Based Shape Items

While the *Add Shape* tool provides way to create simple and predefined geometric item, the *Add Node Item* tool helps you create a custom and more advanced geometric item. For polylines or polygons, you can draw as many lines or sides as you want and vertices of the items can be independently and directly manipulated using the *Edit Nodes Item*. The item itself can be manipulated as exposed in *Interacting with layout items*.

To add a node-based shape:

1. Click the Add Node Item icon
2. Select either Add Polygon or Add Polyline tool
3. Perform consecutive left clicks to add nodes of your item. If you hold down the `Shift` key while drawing a segment, it is constrained to follow an orientation multiple of 45°.
4. When you're done, right-click to terminate the shape.

You can customize the appearance of the shape in the *Item Properties* panel.



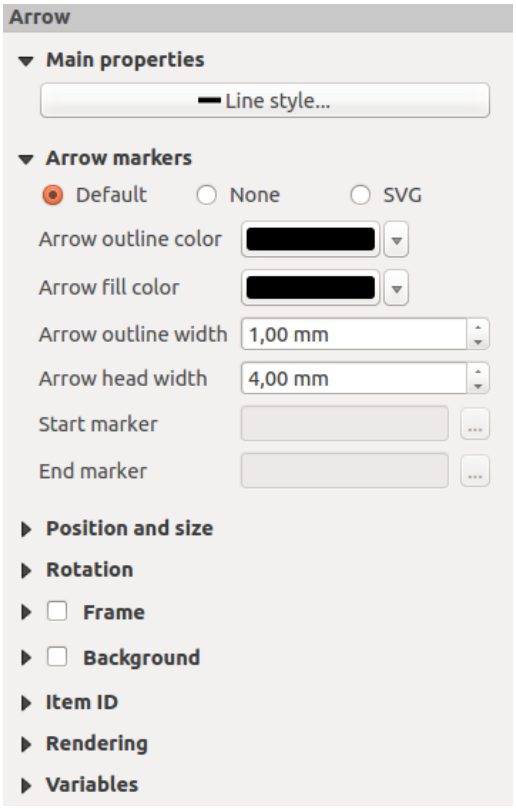
18.62 Fig. : Polygon Node Shape Item Properties Panel

In the *Main properties*, you can set the style of the shape using the advanced *symbol* and *color* selector widget…

For polyline node items, you can also parameterize the *Line markers* i.e. add:

- start and/or end markers with options:
  - *None*: draws a simple polyline.
  - *Arrow*: adds a regular triangular arrow head that you can customize.
  - *SVG* marker: uses an `SVG` file as arrow head of the item.
- customize the arrow head:
  - *Arrow stroke color*: sets the stroke color of the arrow head.
  - *Arrow fill color*: sets the fill color of the arrow head.
  - *Arrow stroke width*: sets the stroke width of the arrow head.
  - *Arrow head width*: sets the size of the arrow head.

SVG images are automatically rotated with the line. Stroke and fill colors of QGIS predefined SVG images can be changed using the corresponding options. Custom SVG may require some tags following this *instruction*.



18.63 Fig. : Polyline Node Shape Item Properties Panel

The Arrow Item

The Add Arrow tool is a shortcut to create an arrow-enabled polyline by default and thus has the same properties and behavior as a *polyline node item*.

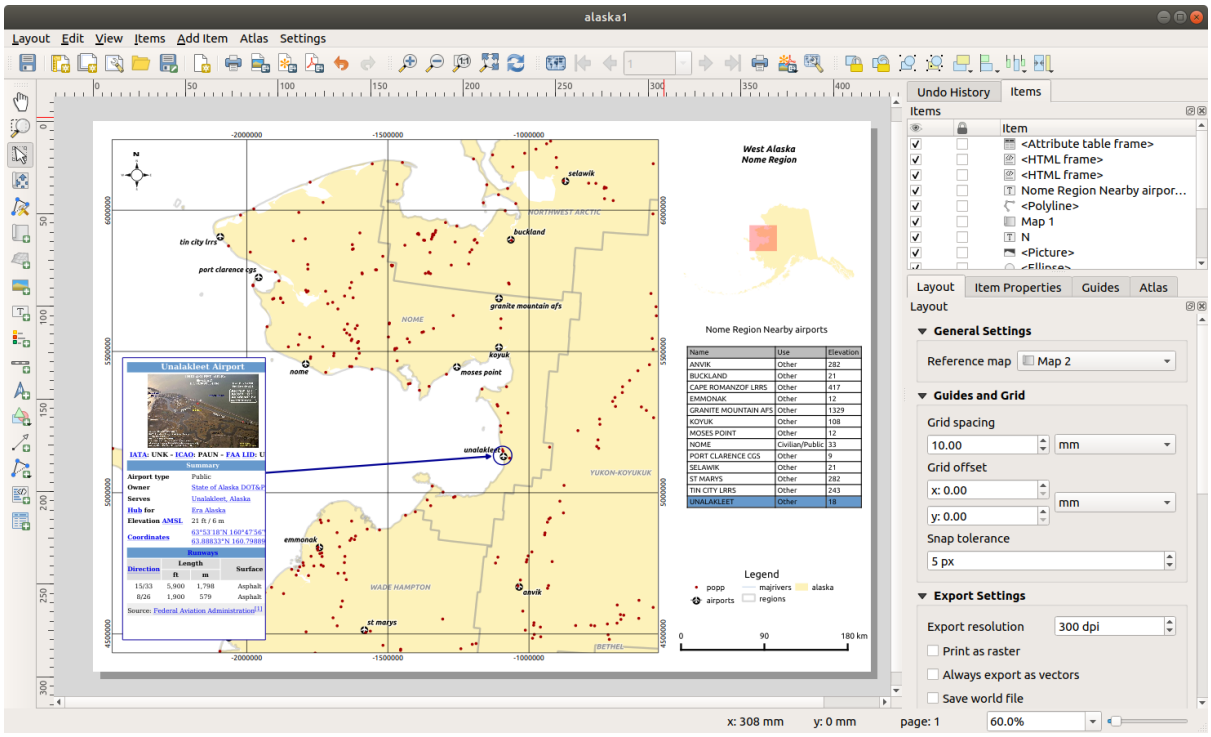
Actually, the arrow item can be used to add a simple arrow, for example, to show the relation between two different print layout items. However, to create a north arrow, the *image item* should be considered first as it gives access to a set of north arrows in `.SVG` format that you can sync with a map item so that it rotates automatically with it.

Editing a node item geometry

A specific tool is provided to edit node-based shapes through Edit Nodes Item. Within this mode, you can select a node by clicking on it (a marker is displayed on the selected node). A selected node can be moved either by dragging it or by using the arrow keys. Moreover, in this mode, you are able to add nodes to an existing shape: double-click on a segment and a node is added at the place you click. Finally, you can remove the currently selected node by hitting the `Del` key.

18.3 Išvesties sukūrimas

Fig. 18.64 rodo spausdinimo išdėstymo pavyzdį su visais ankstesnėje skiltyje aprašytais išdėstymo elementų tipais.



18.64 Fig. : Spausdinimo išdėstymas su pridėtais žemėlapio vaizdu, legenda, piešiniu, masteliu, koordinatėmis, tekstu ir HTML rėmu.

Per meniu *Išdėstymai* arba įrankinę jūs galite išvesti spausdinimo išdėstymą į skirtingų formatų failus. Taipogi galima keisti rezoliuciją (spausdinimo kokybę) ir popieriaus dydį:

- Piktograma Spausdinti leidžia jums spausdinti išdėstymą prijungtame spausdintuve ar į PostScript failą, priklausomai nuo to, kokios įdiegtos spausdinimo tvarkyklės.
- Piktograma Eksportuoti kaip piešinį eksportuoja spausdinimo išdėstymo vaizdą į formatus, tokius kaip `PNG`, `BMP`, `TIF`, `JPG` ir daug kitų…
- Piktograma Eksportuoti kaip SVG leidžia įrašyti spausdinimo išdėstymą kaip `SVG` (kintamo dydžio vektorinė grafika).
- Piktograma Eksportuoti kaip PDF įrašo nurodytą spausdinimo išdėstymo tiesiogiai kaip `PDF` (pernešamo dokumento formato) failą.

18.3.1 Eksporto nustatymai

Whenever you export a print layout, there are a selection of export settings QGIS needs to check in order to produce the most appropriate output. These configurations are:

- The *Export settings* of the *Layout* panel, such as *Export resolution*, *Print as raster Always export as vectors* or *Save world file*
- *Exclude page from exports* in the *page item properties* panel
- *Exclude item from exports* in the *item properties* panel

Moreover, a number of predefined checks are automatically applied to the layout. Currently these checks include testing that scalebars are correctly linked to map items, and that map overview items are also correctly linked to a map. If the checks fail, you are shown a nice warning advising you of the issue.

18.3.2 Export as Image

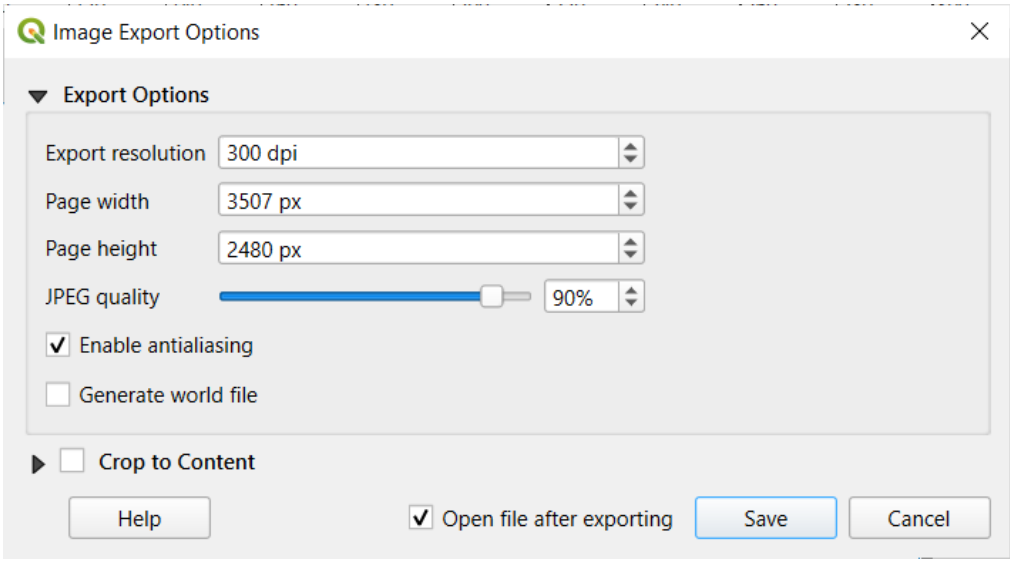
To export a layout as an image:

1. Click the Export as image icon
2. Select the image format, the folder and filename (e.g. `myill.png`) to use. If the layout contains more than one page, each page will be exported to a file with the given filename with the page number appended (e.g. `myill_2.png`).
3. In the next (*Image Export Options*) dialog:
   - You can override the print layout *Export resolution* and the exported page dimensions (as set in *Layout* panel).
   - For JPEG and JPG image formats, you can set the *JPEG Quality* of the output image.
   - Image rendering can also be improved with the *Enable antialiasing* option.
   - If you want to export your layout as a **georeferenced image** (e.g., to share with other projects), check the *Generate world file* option, and an *ESRI World File* with the same name as the exported image, but a different extension (`.tfw` for TIFF, `.pnw` for PNG, `jgw` for JPEG, …) will be created when exporting. This option can also be checked by default in the *layout panel*.

     > **Pastaba**
     >
     > For multi-page output, only the page that contains the *reference map* will get a world file (assuming that the *Generate world file* option is checked).

   - By checking *Crop to content* option, the image output by the layout will include the minimal area enclosing all the items (map, legend, scale bar, shapes, label, image…) of each page of the composition:
     - If the composition includes a single page, then the output is resized to include EVERYTHING on the composition. The page can then be reduced or extended to all items depending on their position (on, above, below, left or right of the page).
     - In case of a multi-page layout, each page will be resized to include items in its area (left and right sides for all pages, plus top for the first page and bottom for the last page). Each resized page is exported to a separate file.

     The *Crop to content* dialog also lets you add margins around the cropped bounds.
   - By checking *Open file after exporting* the exported file will automatically open in the default image viewer.



18.65 Fig. : Image Export Options

Patarimas

Use image formats that support transparency when items extend beyond the paper extent

Layout items may be placed outside the paper extent. When exporting with the *Crop to content* option, the resulting image may therefore extend beyond the paper extent. Since the background outside of the paper extent will be transparent, for image formats that do not support transparency (e.g. `BMP` and `JPG`) the transparent background will be rendered as full black, „corrupting“ the image. Use transparency-compatible formats (e.g. `TIFF` and `PNG`) in such cases.

Pastaba

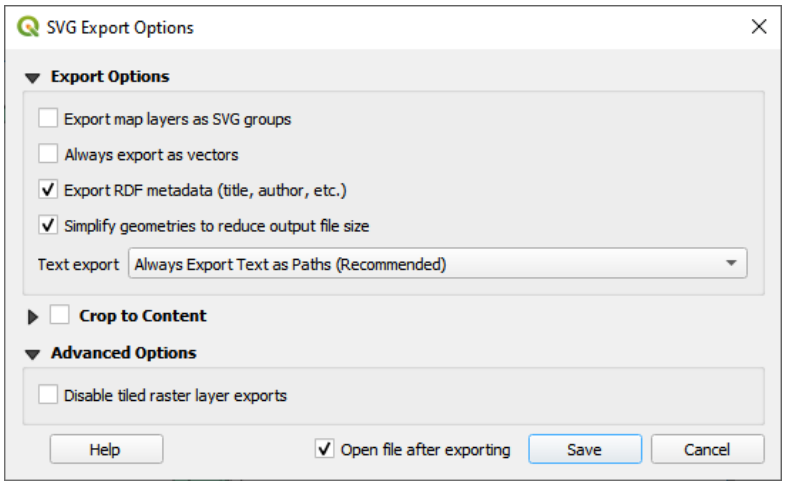
When supported by the format (e.g. `PNG`) and the underlying Qt library, the exported image may include *project metadata* (author, title, date, description…)

18.3.3 Export as SVG

To export a layout as SVG:

1. Click the Export as SVG icon
2. Fill in the path and filename (used as a base name for all the files in case of multi-page composition, as for image export)
3. In the next *SVG Export Options* dialog, you can override the layout default *export settings* or configure new ones:
   - *Export map layers as SVG groups*: exported items are grouped within layers whose name matches the layer names from QGIS, making it much easier to understand the contents of the document.
   - *Always export as vectors*: some rendering options require items to be rasterized for a better rendering. Check this option to keep the objects as vectors with the risk that the appearance of the output file may not match the print layout preview (for more details, see *Eksporto nustatymai*).
   - *Eksportuoti RDF metaduomenis* - pridedami dokumento duomenys, tokie kaip pavadinimas, autorius, data, aprašymas…

- *Simplify geometries to reduce output file size*: this avoids exporting ALL geometry vertices, which can result in a ridiculously complex and large export file size that could fail to load in other applications. Geometries will be simplified while exporting the layout in order to remove any redundant vertices which are not discernably different at the export resolution (e.g. if the export resolution is `300 dpi`, vertices that are less than `1/600 inch` apart will be removed).
- Nustatymas *Teksto eksportas*: valdo ar teksto užrašai visada ar pageidaujamai eksportuojami kaip *teksto ar kontūro objektai*.
- Apply *Crop to content* *option*
- *Disable tiled raster layer exports*: When exporting files, QGIS uses a built-in raster layer tiled rendering that saves memory. Sometimes, this can cause visible „seams“ in the rasters for generated files. Checking this option would fix that, at the cost of a higher memory usage during exports.
- By checking *Open file after exporting* the exported file will automatically open in the default SVG viewer.



18.66 Fig. : SVG Export Options

Pastaba

Currently, the SVG output is very basic. This is not a QGIS problem, but a problem with the underlying Qt library. This will hopefully be sorted out in future versions.

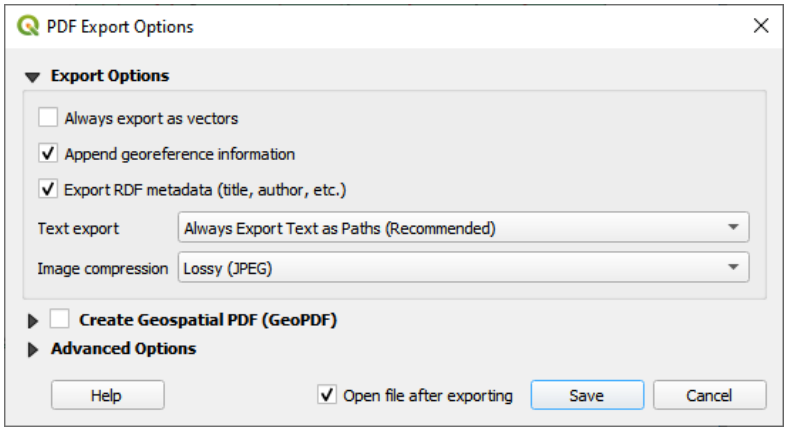
18.3.4 Export as PDF

To export a layout as PDF:

1. Click the Export as PDF icon
2. Fill in the path and filename: unlike for image and SVG export, all the pages in the layout are exported to a single PDF file.
3. In the next *PDF Export Options* dialog, you can override the layout default *export settings* or configure new ones:
   - *Always export as vectors*: some rendering options require items to be rasterized for a better rendering. Check this option to keep the objects as vectors with the risk that the appearance of the output file may not match the print layout preview (for more details, see *Eksporto nustatymai*).

> **Pastaba**
>
> **Mind opacity settings for vector layers**
>
> When you set a layer-wide opacity (or opacity for any vector symbology) to less than 100%, it forces a rasterized rendering of those objects during PDF exports. Depending on the size of the layer, this may considerably increase the PDF file size.

- *Append georeference information*: available only if the *reference map*, from which the information is taken, is on the first page.
- *Eksportuoti RDF metaduomenis* - pridedami dokumento duomenys, tokie kaip pavadinimas, autorius, data, aprašymas…
- Nustatymas *Teksto eksportas*: valdo ar teksto užrašai visada ar pageidaujamai eksportuojami kaip *teksto ar kontūro objektai*.
- Control the PDF *Image compression* using:
  - *Lossy (JPEG)*, which is the default compression mode
  - or *Lossless*, which creates bigger files in most cases, but is much more suitable for printing outputs or for post-production in external applications (requires Qt 5.13 or later).
- *Create Geospatial PDF*: Generate a georeferenced PDF file.
- *Disable tiled raster layer exports*: When exporting files, QGIS uses tiled based rendering that saves memory. Sometimes, this can cause visible „seams“ in the rasters for generated files. Checking this option would fix that, at the cost of a higher memory usage during exports.
- *Simplify geometries to reduce output file size*: Geometries will be simplified while exporting the layout by removing vertices that are not discernably different at the export resolution (e.g. if the export resolution is `300 dpi`, vertices that are less than `1/600 inch` apart will be removed). This can reduce the size and complexity of the export file (very large files can fail to load in other applications).
- By checking *Open file after exporting* the exported file will automatically open in the default PDF viewer.



18.67 Fig. : PDF Export Options

> **Pastaba**
>
> Geospatial PDF export is supported, and a number of Geospatial PDF specific options are available:
>
> - *Format* (Geospatial PDF format - there are some variations),

- *Include multiple map themes* (specify map themes to include),
- *Include vector feature information* (choose the layers and group them into logical PDF groups).

Pastaba

Exporting a print layout to formats that supports georeferencing (e.g. `PDF` and `TIFF`) creates a georeferenced output by default.

Pastaba

When exporting maps or layouts to vector formats such as PDF or SVG, line pattern fills are rendered using a line-by-line vector-based approach instead of rasterized tiles. This change results in smaller and higher-quality output files, with individual pattern lines remaining fully editable in external vector graphics software.

18.3.5 Generate an Atlas

Atlas functions allow you to create map books in an automated way. Atlas uses the features of a table or vector layer (*Coverage layer*) to create an output for each feature (**atlas feature**) in the table / layer. The most common usage is to zoom a map item to the current atlas feature. Further use cases include:

- a map item showing, for another layer, only features that share the same attribute as the atlas feature or are within its geometry.
- a label or HTML item whose text is replaced as features are iterated over
- a table item showing attributes of associated *parent or children* features of the current atlas feature…

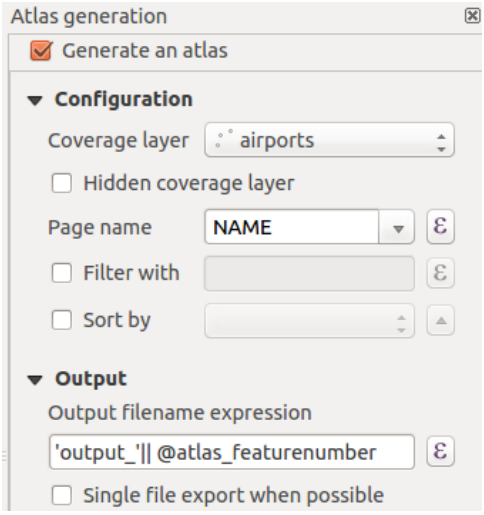
For each feature, the output is processed for all pages and items according to their exports settings.

Patarimas

Use variables for more flexibility

QGIS provides a large panel of functions and *variables*, including atlas related ones, that you can use to manipulate the layout items, but also the symbology of the layers, according to atlas status. Combining these features gives you a lot of flexibility and helps you easily produce advanced maps.

To enable the generation of an atlas and access atlas parameters, refer to the *Atlas* panel. This panel contains the following (see Fig. 18.68):



18.68 Fig. : Atlas Panel

- *Generate an atlas* enables or disables atlas generation.
- *Configuration*
  - A *Coverage layer* combo box that allows you to choose the table or vector layer containing the features to iterate over.
  - An optional *Hidden coverage layer* that, if checked, will hide the coverage layer (but not the other layers) during the generation.
  - An optional *Page name* combo box to specify the name for the feature page(s). You can select a field of the coverage layer or set an *expression*. If this option is empty, QGIS will use an internal ID, according to the filter and/or the sort order applied to the layer.
  - An optional *Filter with* text area that allows you to specify an expression for filtering features from the coverage layer. If the expression is not empty, only features that evaluate to `True` will be processed.
  - An optional *Sort by* that allows you to sort features of the coverage layer (and the output), using a field of the coverage layer or an expression. The sort order (either ascending or descending) is set by the two-state *Sort direction* button that displays an up or a down arrow.
- *Output* - this is where the output of the atlas can be configured:
  - An *Output filename expression* textbox that is used to generate a filename for each atlas feature. It is based on expressions. is meaningful only for rendering to multiple files.
  - A *Single file export when possible* that allows you to force the generation of a single file if this is possible with the chosen output format (`PDF`, for instance). If this field is checked, the value of the *Output filename expression* field is meaningless.
  - An *Image export format* drop-down list to select the output format when using the Export atlas as Images… button.

Control map by atlas

The most common usage of atlas is with the map item, zooming to the current atlas feature, as iteration goes over the coverage layer. This behavior is set in the *Controlled by atlas* group properties of the map item. See *Controlled by atlas* for different settings you can apply on the map item.

Customize labels with expression

In order to adapt labels to the feature the atlas iterates over, you can include expressions. Make sure that you place the expression part (including functions, fields or variables) between `[%` and `%]` (see *The Label Item* for more details).

For example, for a city layer with fields `CITY_NAME` and `ZIPCODE`, you could insert this:

```
The area of [% concat( upper(CITY_NAME), ',', ZIPCODE, ' is ',
format_number($area/1000000, 2) ) %] km2
```

or, another combination:

```
The area of [% upper(CITY_NAME)%],[%ZIPCODE%] is
[%format_number($area/1000000,2) %] km2
```

The information `[% concat( upper(CITY_NAME), ',', ZIPCODE, ' is ', format_number($area/1000000, 2) ) %]` is an expression used inside the label. Both expressions would result in the following type of label in the generated atlas:

```
The area of PARIS,75001 is 1.94 km2
```

Explore Data-defined override buttons with atlas

There are several places where you can use a Data defined override button to override the selected setting. This is particularly useful with atlas generation. See *Nuo duomenų priklausančių permušimų nustatymai* for more details on this widget.

For the following examples the `Regions` layer of the QGIS sample dataset is used and selected as *Coverage layer* for the atlas generation. We assume that it is a single page layout containing a map item and a label item.

When the height (north-south) of a region extent is greater than its width (east-west), you should use *Portrait* instead of *Landscape* orientation to optimize the use of paper. With a Data Defined Override button you can dynamically set the paper orientation.

Right-click on the page and select *Page Properties* to open the panel. We want to set the orientation dynamically, using an expression depending on the region geometry, so press the button of field *Orientation*, select *Edit…* to open the *Expression string builder* dialog and enter the following expression:

```
CASE WHEN bounds_width(@atlas_geometry) > bounds_height(@atlas_geometry)
THEN 'Landscape' ELSE 'Portrait' END
```

Now if you *preview the atlas*, the paper orients itself automatically, but item placements may not be ideal. For each Region you need to reposition the location of the layout items as well. For the map item you can use the button of its *Width* property to set it dynamic using the following expression:

```
@layout_pagewidth - 20
```

Likewise, use the button of the *Height* property to provide the following expression to constrain map item size:

```
@layout_pageheight - 20
```

To ensure the map item is centered in the page, set its *Reference point* to the upper left radio button and enter `10` for its *X* and *Y* positions.

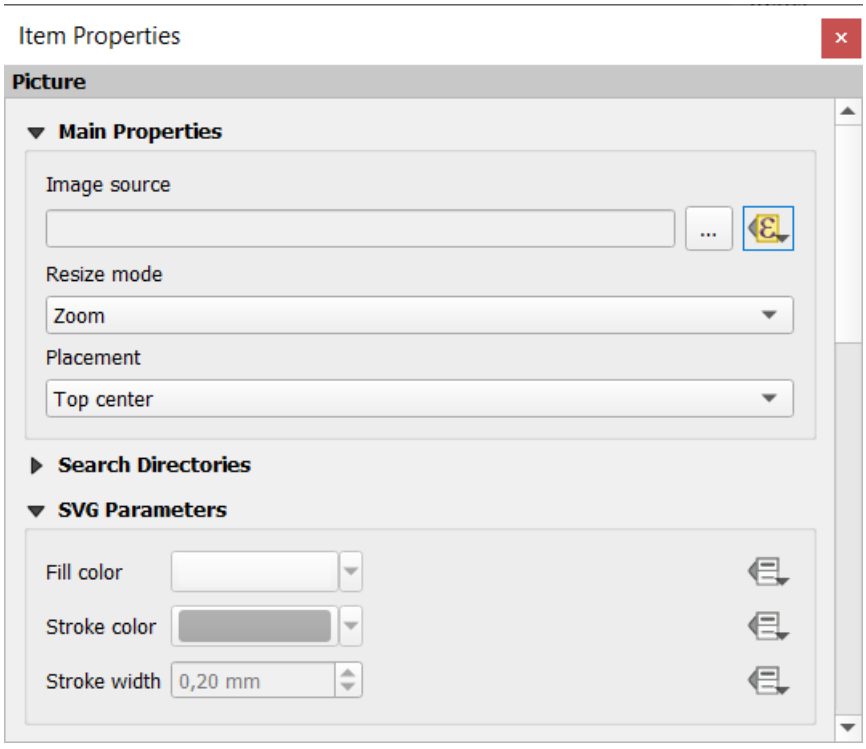
Let's add a title above the map in the center of the page. Select the label item and set the horizontal alignment to *Center*. Next move the label to the right position, choose the middle button for the *Reference point*, and provide the following expression for field *X*:

```
@layout_pagewidth / 2
```

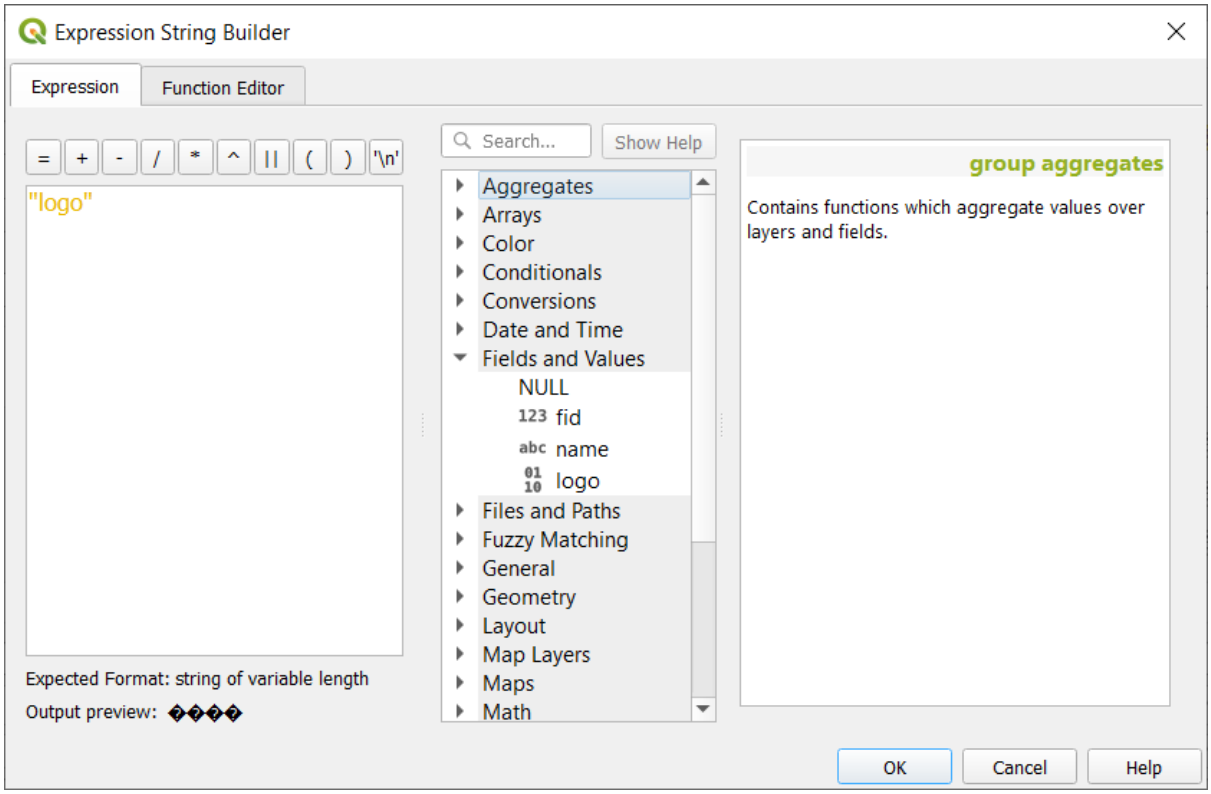
For all other layout items you can set the position in a similar way so they are correctly positioned both for portrait and landscape. You can also do more tweaks such as customizing the title with feature attributes (see *Customize labels with expression* example), changing images, resizing the number of legend columns number according to page orientation, …

The information provided here is an update of the excellent blog (in English and Portuguese) on the Data Defined Override options Multiple_format_map_series_using_QGIS_2.6 .

Another example for using data-defined override buttons is the usage of a dynamic picture. For the following examples we use a geopackage layer containing a BLOB field called `logo` with the field type binary (see *Naujo GeoPackage sluoksnio kūrimas*). For every feature there is defined a different picture so that the atlas can iterate over as described in *Preview and generate an atlas*. All you need to do is add a picture in the print layout and go to its *Item properties* in the atlas context. There you can find a data-defined override button in the *Image source* section of the *Main Properties*.



In the following window choose *Edit* so that the *Expression String Builder* opens. From the *Fields and values* section you can find the BLOB field that was defined in the geopackage layer. Double-click the field name `logo` and click *OK*.



The atlas iterates over the entries in the BLOB field provided that you choose the geopackage layer as *Coverage layer* (further instructions you can find in *Preview and generate an atlas*).

These are just two examples of how you can use some advanced settings with atlas.

Preview and generate an atlas

18.69 Fig. : Atlas Preview toolbar

Once the atlas settings have been configured, and layout items (map, table, image…) linked to it, you can create a preview of all the pages by choosing *Atlas* ► *Preview Atlas* or clicking the Preview Atlas icon. You can then use the arrows to navigate through all the features:

- First feature
- Previous feature
- Next feature
- Last feature

You can also use the combo box to select and preview a specific feature. The combo box shows atlas feature names according to the expression set in the atlas *Page name* option.

As for simple compositions, an atlas can be generated in different ways (see *Išvesties sukūrimas* for more information - just use tools from the *Atlas* menu or toolbar instead of the *Layout* menu.

This means that you can directly print your compositions with *Atlas* ► *Print Atlas*. You can also create a PDF using *Atlas* ► *Export Atlas as PDF…*: You will be asked for a directory to save all the generated PDF files, except if the *Single file export when possible* has been selected. In that case, you'll be prompted to give a filename.

With *Atlas* ► *Export Atlas as Images…* or *Atlas* ► *Export Atlas as SVG…* tool, you're also prompted to select a folder. Each page of each atlas feature composition is exported to the image file format set in *Atlas* panel or to SVG.

Pastaba

With multi-page output, an atlas behaves like a layout in that only the page that contains the *General settings* will get a world file (for each feature output).

Patarimas

Print a specific atlas feature

If you want to print or export the composition of only one feature of the atlas, simply start the preview, select the desired feature in the drop-down list and click on *Layout* ► *Print* (or *Export…* to any supported file format).

Use project defined relations for atlas creation

For users with HTML and Javascript knowledge it is possible to operate on GeoJSON objects and use project defined relations from the QGIS project. The difference between this approach and using expressions directly inserted into the HTML is that it gives you a full, unstructured GeoJSON feature to work with. This means that you can use existing Javascript libraries and functions that operate on GeoJSON feature representations.

The following code includes all related child features from the defined relation. Using the JavaScript `setFeature` function it allows you to make flexible HTML which represents relations in whatever format you like (lists, tables, etc). In the code sample, we create a dynamic bullet list of the related child features.

```
// Declare the two HTML div elements we will use for the parent feature id
// and information about the children
<div id="parent"></div>
<div id="my_children"></div>

<script type="text/javascript">
   function setFeature(feature)
   {
     // Show the parent feature's identifier (using its "ID" field)
     document.getElementById('parent').innerHTML = feature.properties.ID;
     //clear the existing relation contents
     document.getElementById('my_children').innerHTML = '';
     feature.properties.my_relation.forEach(function(child_feature) {
     // for each related child feature, create a list element
     // with the feature's name (using its "NAME" field)
       var node = document.createElement("li");
       node.appendChild(document.createTextNode(child_feature.NAME));
       document.getElementById('my_children').appendChild(node);
     });
   }
</script>
```

During atlas creation there will be an iteration over the coverage layer containing the parent features. On each page, you will see a bullet list of the related child features following the parent's identifier.

18.4 Creating a Report

This section will help you set up a report in QGIS.

18.4.1 What is it?

By definition, a GIS report is a document containing information organized in a narrative way, containing maps, text, graphics, tables, etc. A report can be prepared ad hoc, periodic, recurring, regular, or as required. Reports may refer to specific periods, events, occurrences, subjects or locations.

In QGIS, a *Report* is an extension of a *Layouts*.

Reports allow users to output their GIS projects in a simple, quick and structured way.

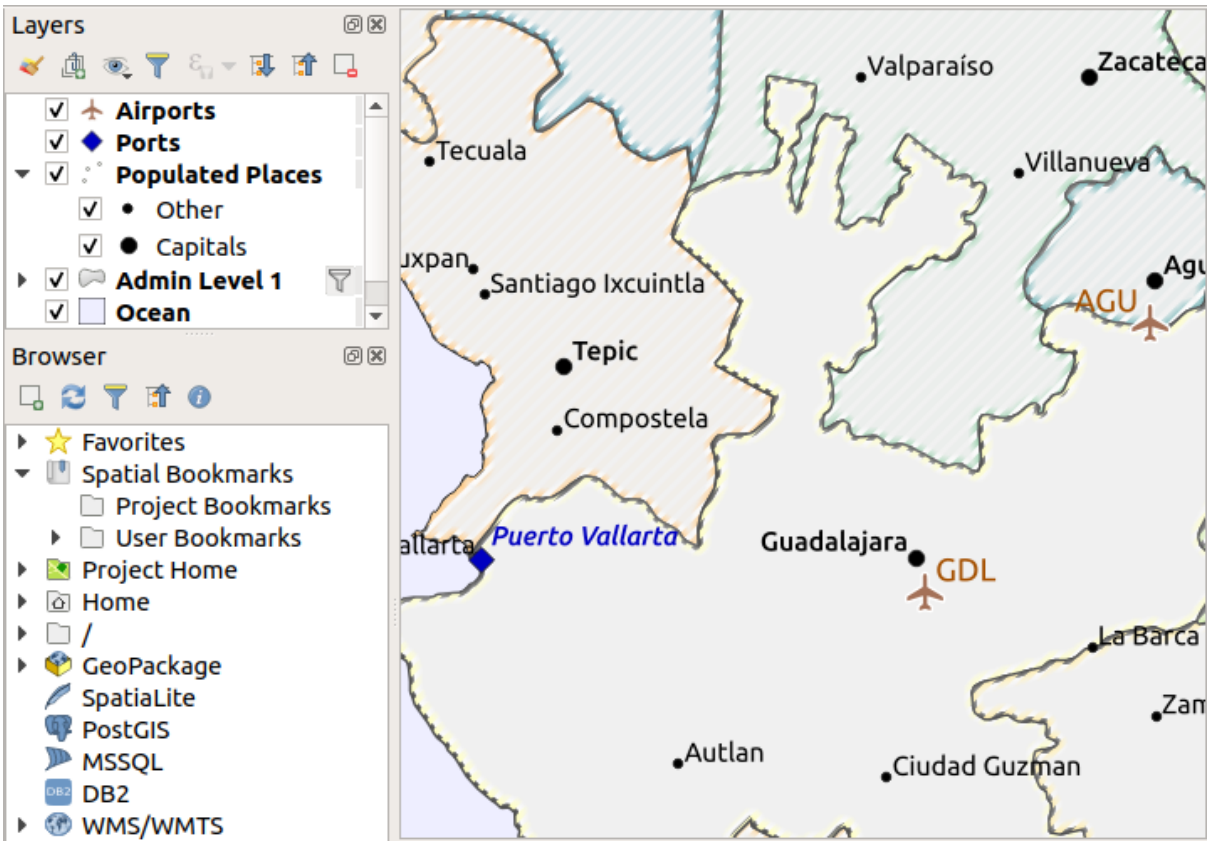
A report can be created with *Project* ► *New Report* or inside the *Project* ► *Layout Manager*.

> **Pastaba**
>
> The maps in QGIS reports behave in the same way as maps in print layouts and atlases. We will concentrate on the specifics of QGIS reports. For details on map handling, see the sections on *print layouts* and *atlases*.

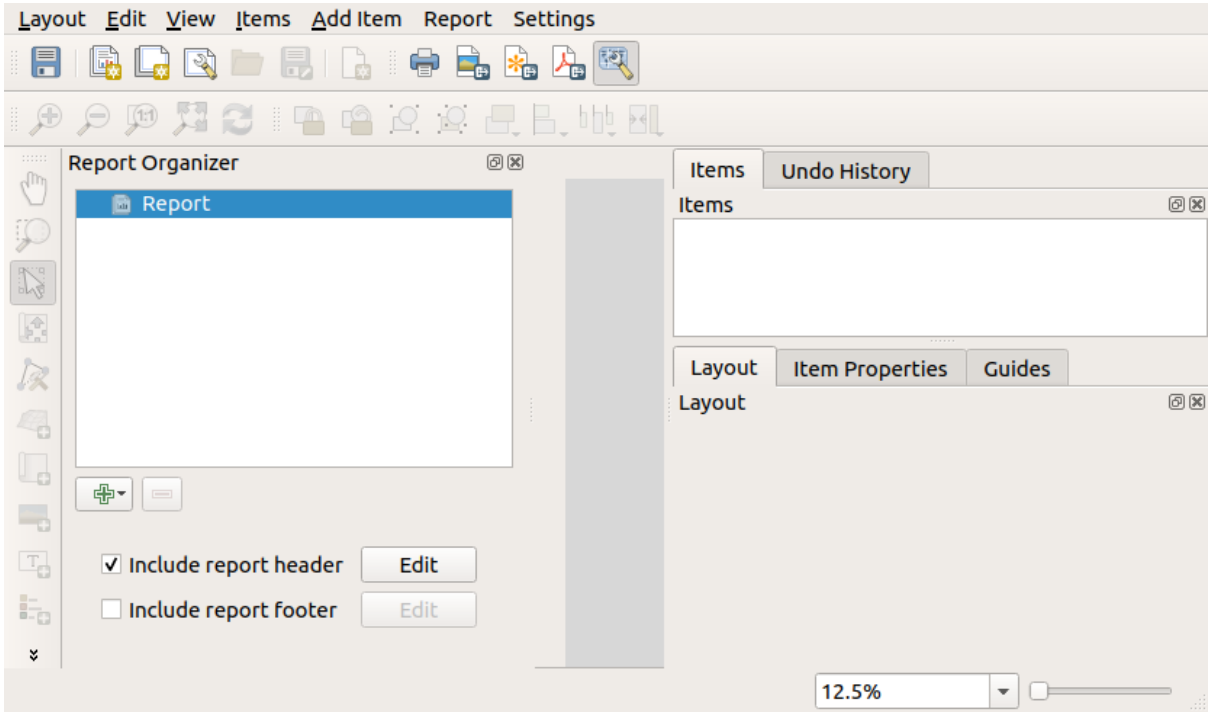
18.4.2 Get started

In the *Layout Manager* dialog a report can be created through *New from template* by selecting the dropdown option *Empty Report* and hitting the *Create…* button.

For this example, we use some administrative boundaries, populated places, ports and airports from the Natural Earth dataset (1:10M).

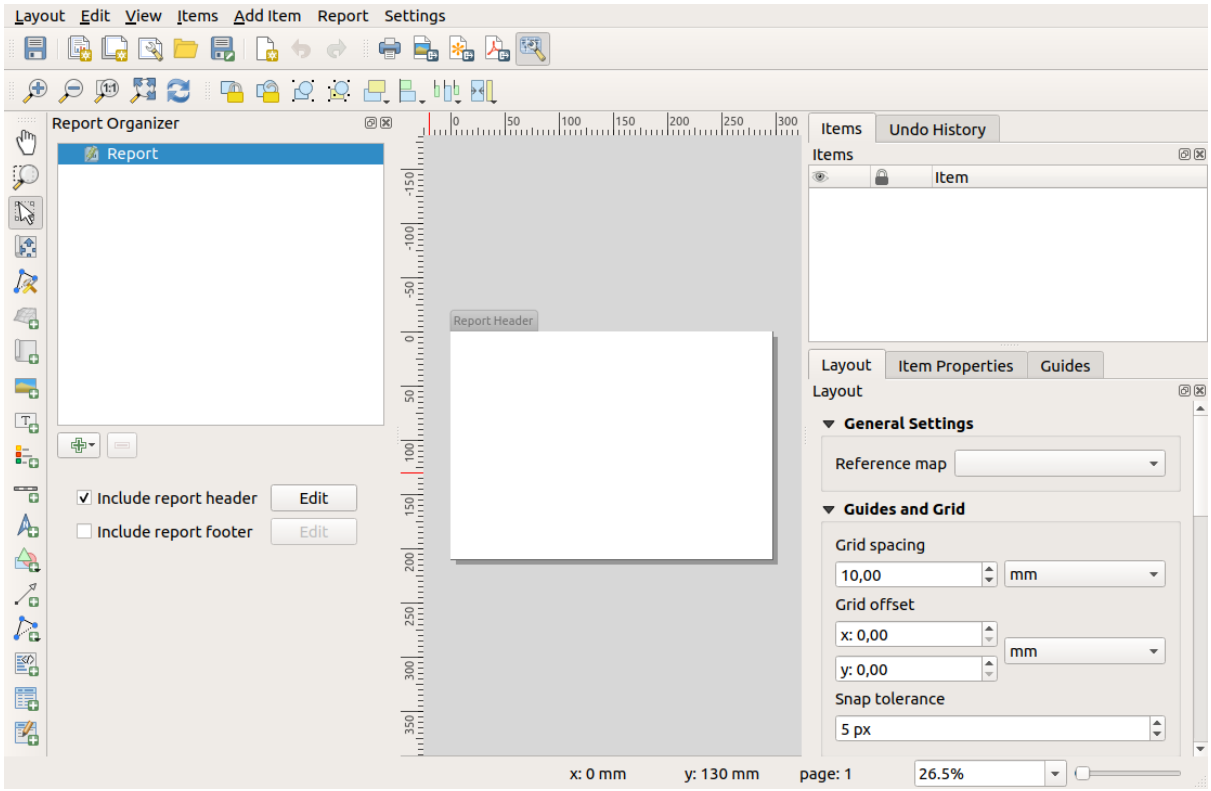


Using the *Project* ► *New Report* command, we create a blank report. Initially, there is not much to look at – the dialog which is displayed looks much like the print layout designer, except for the *Report Organizer* panel to the left:



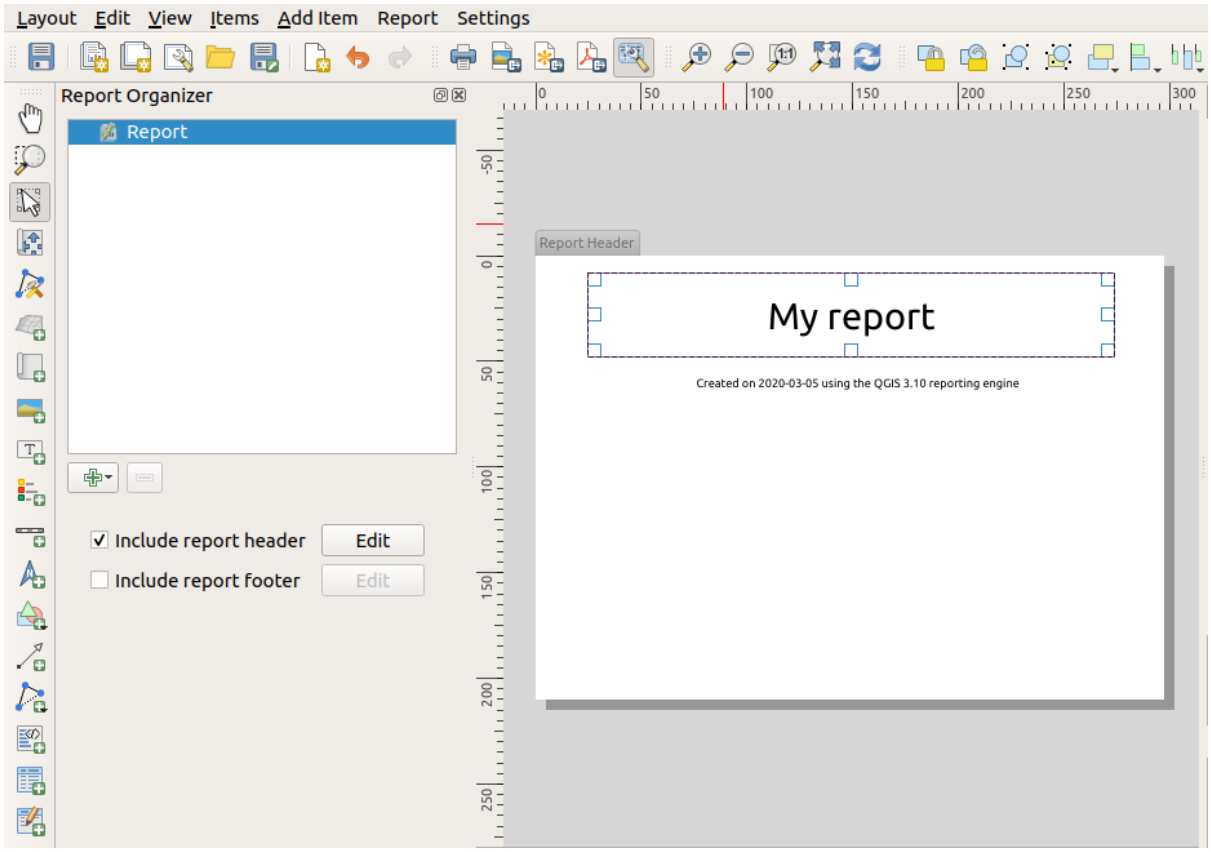
18.4.3 Layout Report Workspace

QGIS reports can consist of multiple, nested sections. In our new blank report we initially only have the main report section. The only options for this report section is *Include report header* and *Include report footer*. If we enable these options, a header will be included as the first page(s) (individual parts of reports can be multi-page if desired) in the report, and a footer will constitute the last page(s). Enable the header (*Include report header*), and hit the *Edit* button next to it:

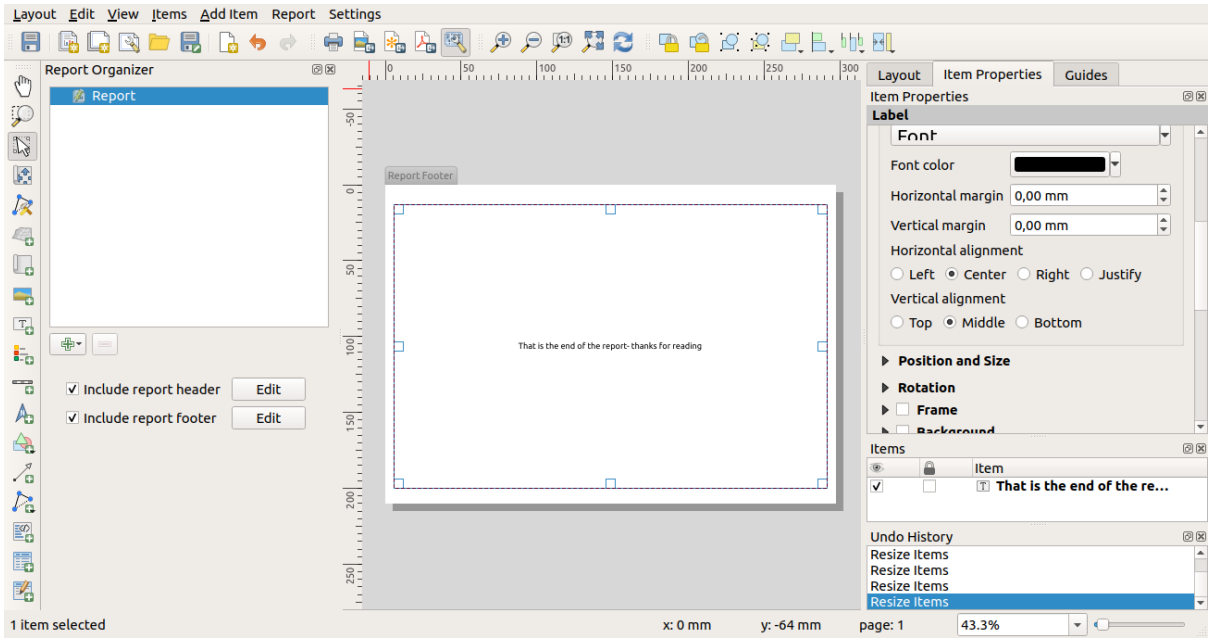


A few things happen as a result. Firstly, an edit pencil is shown next to *Report* in the *Report Organizer*, indicating that the report section is currently being edited in the designer. We also see a new page with a small *Report Header* title. The page has *landscape* orientation by default, but this (and other properties of the page) can be changed by right-clicking on the page and choosing *Page properties*. This will bring up the *Item properties* tab for the page, and page *Size*, *Width*, *Height*, and more can be specified.

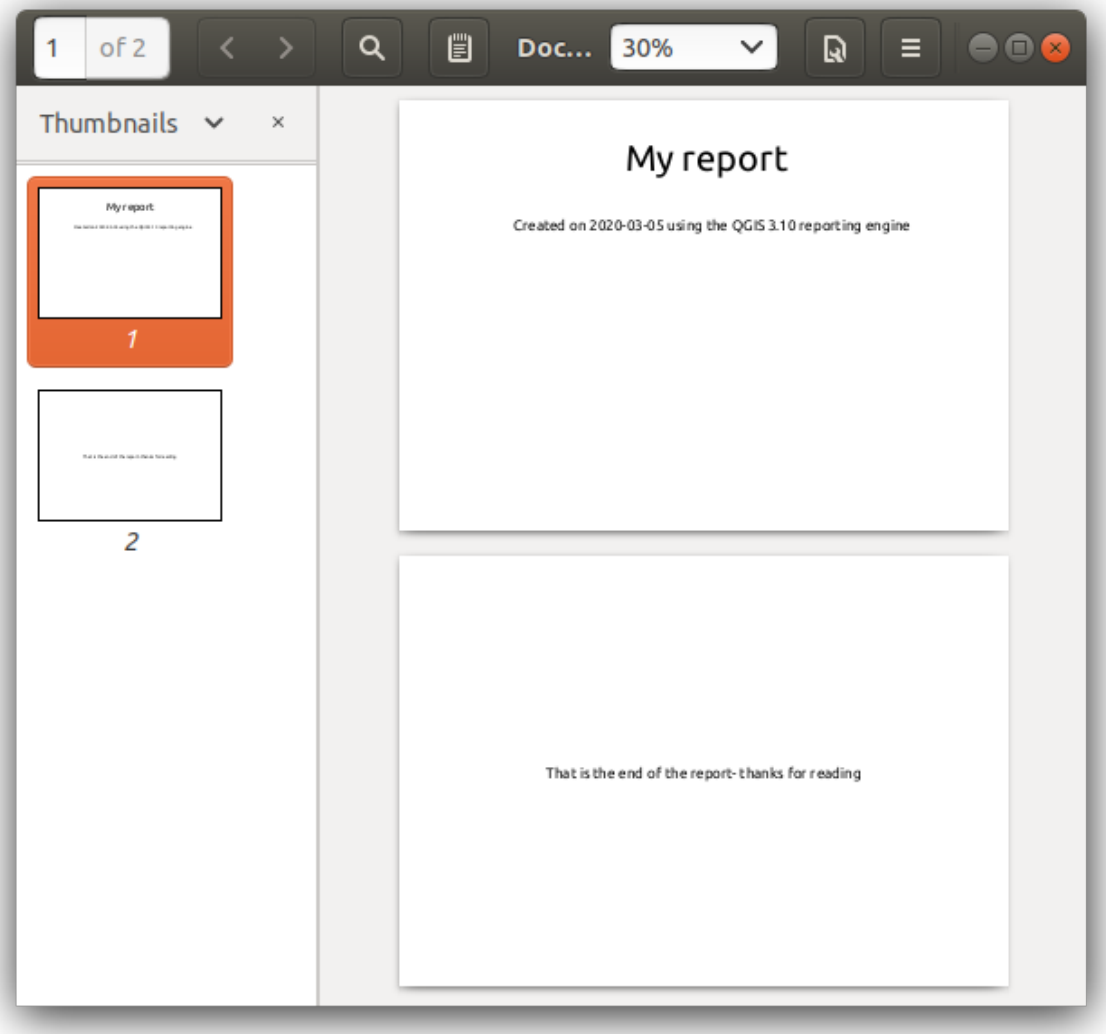
In QGIS reports, every component of the report is made up of individual layouts. They can be created and modified using the same tools as for standard print layouts – so you can use any desired combination of labels, pictures, maps, tables, etc. Let us add some items to our report header to demonstrate:



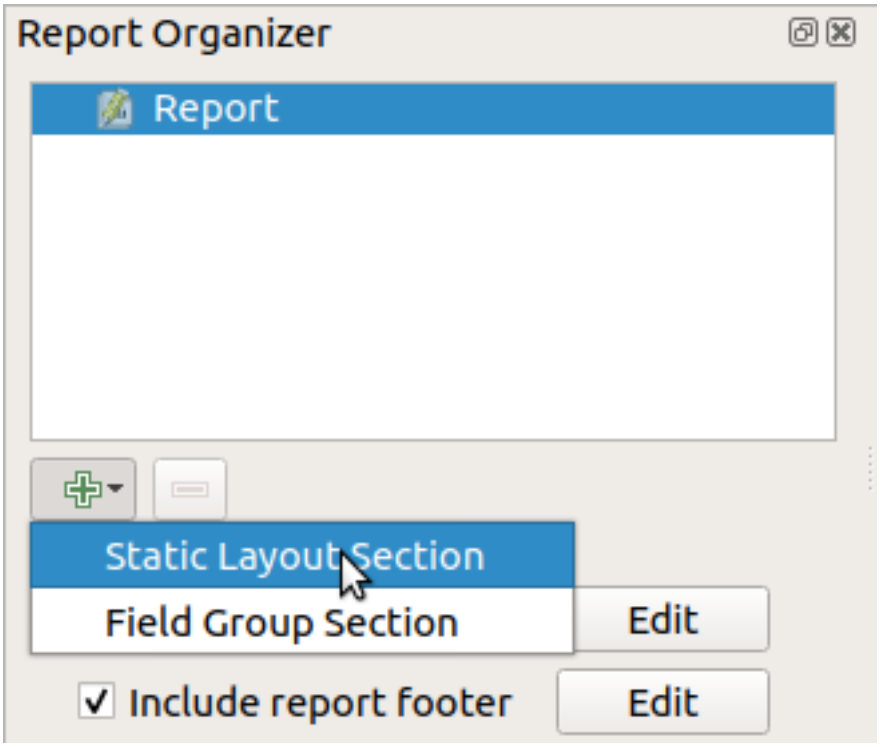
We will also create a simple footer for the report by checking the *Include report footer* option and hitting *Edit*.



Before proceeding further, let us export this report and see what we get. Exporting is done from the *Report* menu – in this case we select *Export Report as PDF…* to render the whole report to a PDF file. Here is the not-very-impressive result – a two page PDF consisting of our header and footer:



Let us make things more interesting. By hitting the Add Section button in the *Report Organizer*, we are given a choice of new sections to add to our report.

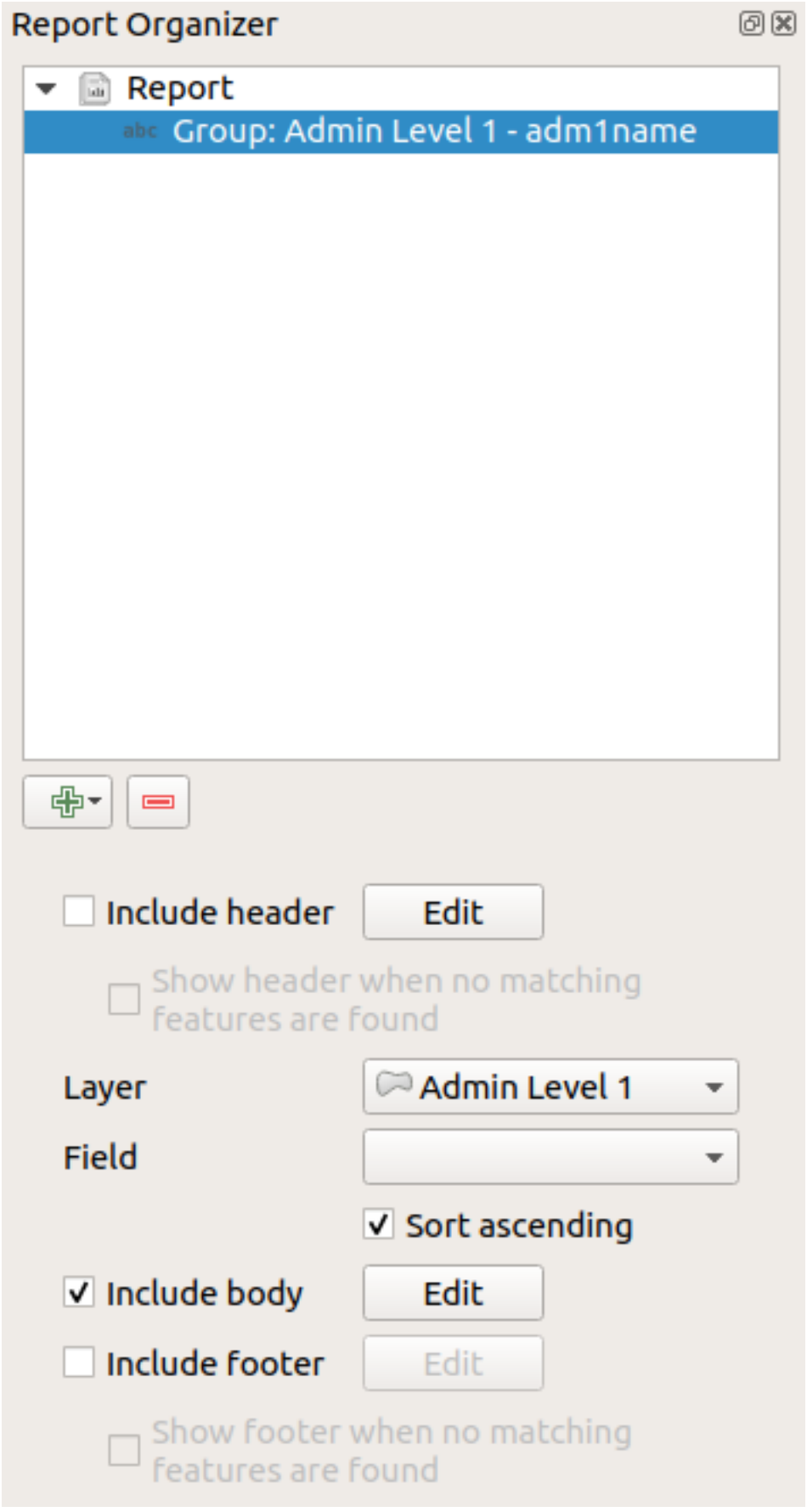


There are two options: *Static Layout Section* and *Field Group Section*.

The *Add Static Layout Section* is a single, static body layout. This can be used to embed static layouts mid-way through a report.

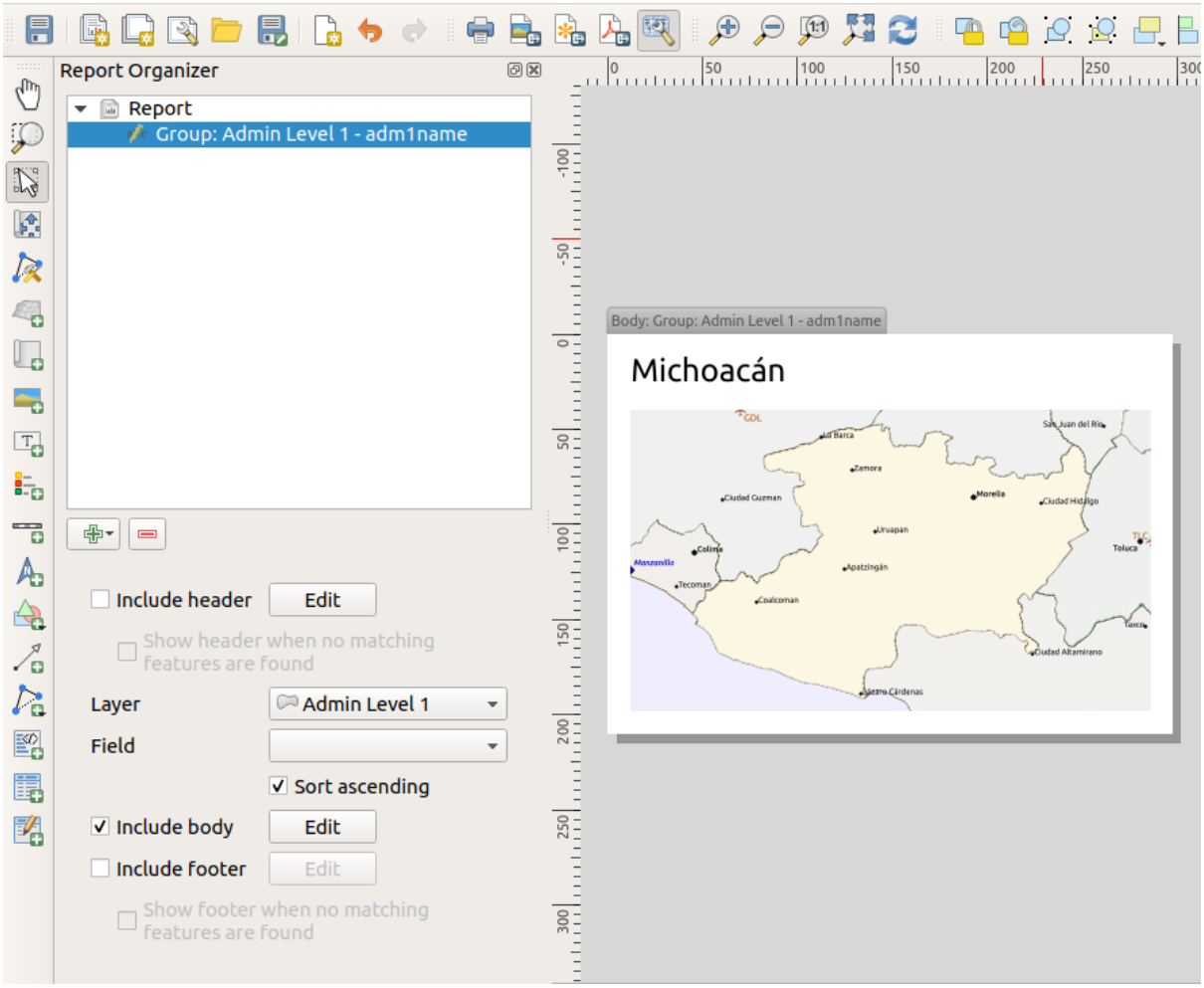
The *Field Group Section* repeats its body layout for every feature of a layer. The features are sorted by the selected grouping feature (with an option for ascending/descending sort). If a field group section has child sections (e.g. another field group section with a different field), then only features with unique values for the group feature are iterated over. This allows nested reports.

For now we will add a *Field Group Section* to our report. At its most basic level, you can think of a *Field Group Section* as the equivalent of a *print atlas*: you select a layer to iterate over, and the report will insert a section for each feature found. Selecting the new *Field Group Section* reveals a number of new related settings:



In this case we've setup our Field Group so that we iterate over all the states from the *Admin Level 1* layer, using the

values from the *adm1name* field. The same options to include header and footer are present, together with a new option to include a *body* for this section. We'll do that, and edit the body:

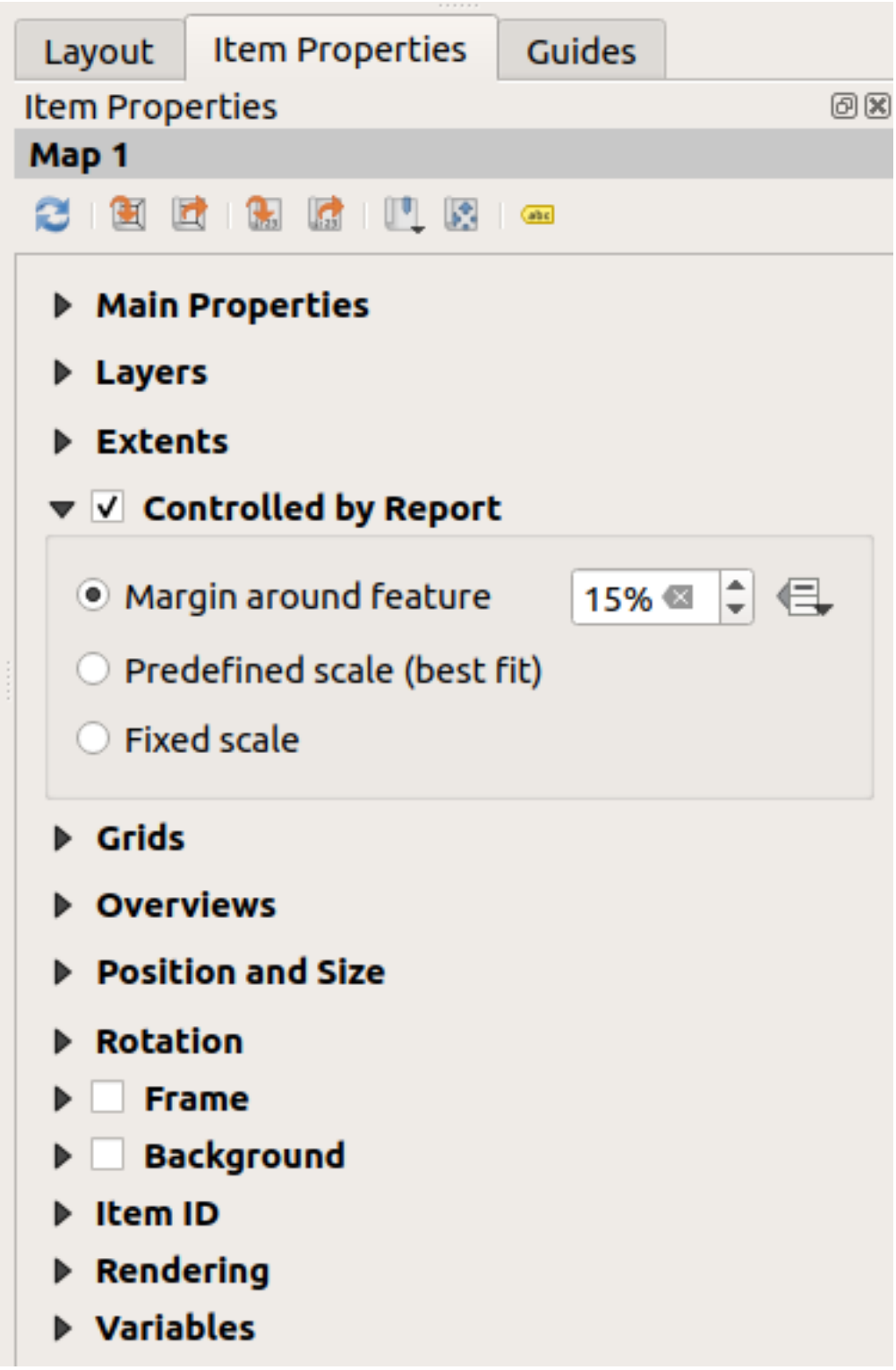


Our body now consists of a map and a label showing the name of the state. To include the name of the state, we selected *Add Item* ► *Add Label* and data defined the text under *Main Properties* with the help of *Insert or Edit an Expression*….

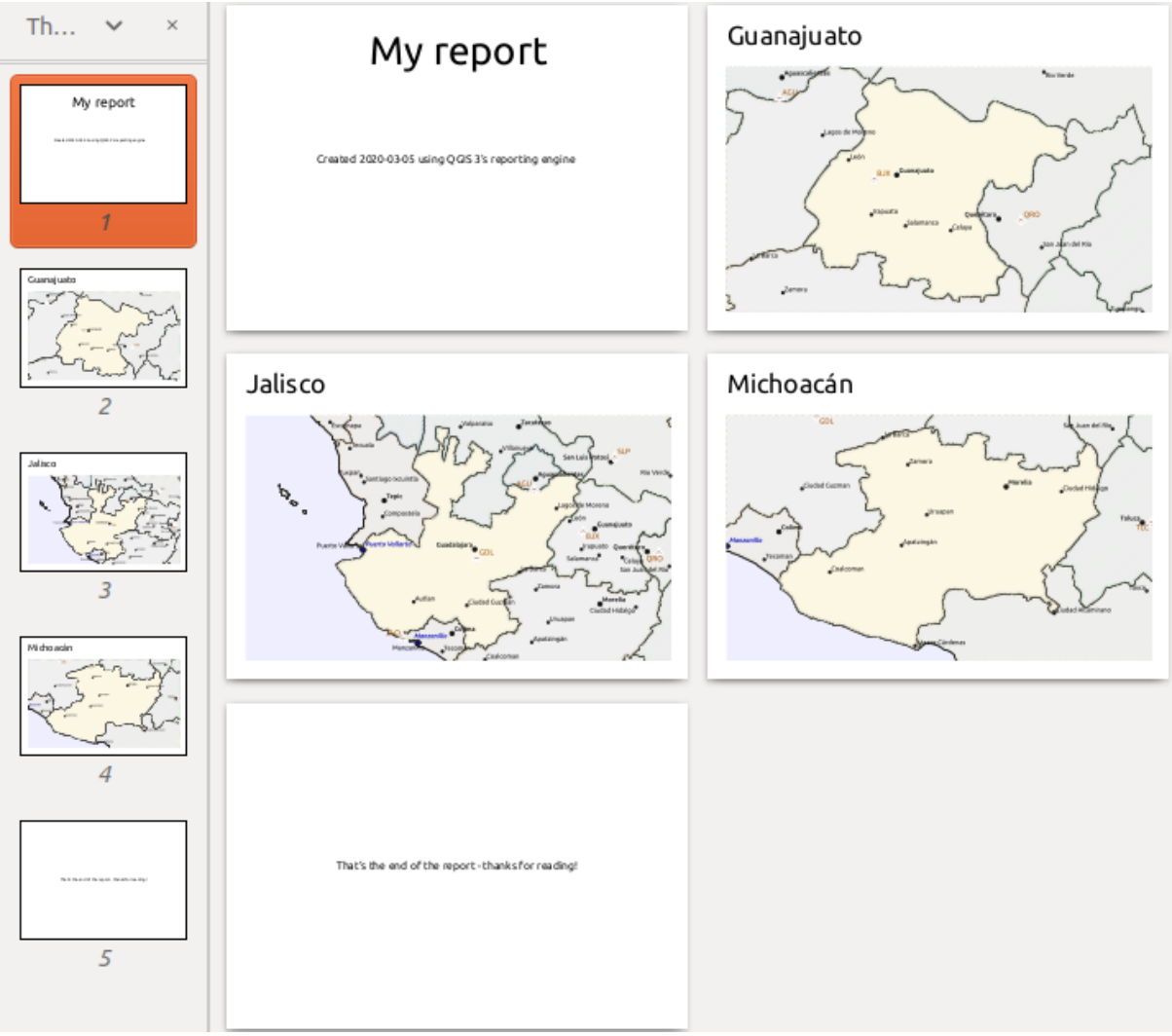
The result was the following expression (*name* is the name of the attribute in the *Admin Level 1* layer that contains the name of the state):

```
[% "name" %]
```

The map is set to follow the current report feature (enabled by checking *Controlled by Report* – just like a map item in an atlas will follow the current atlas feature when *Controlled by Atlas* is checked):



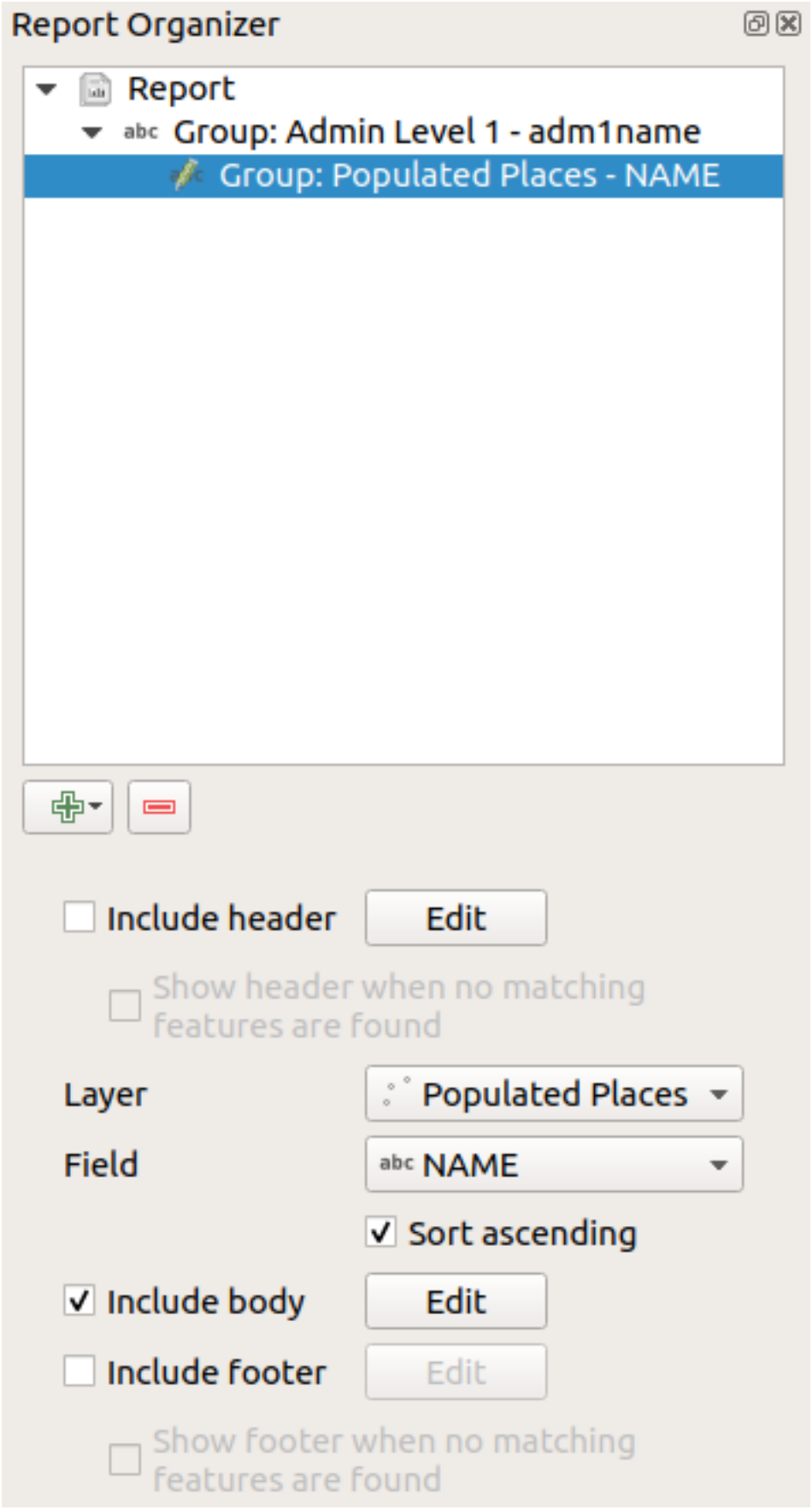
If we went ahead and exported our report now, we'd get something like this:



18.70 Fig. : The report header, a page for each state, and the report footer.

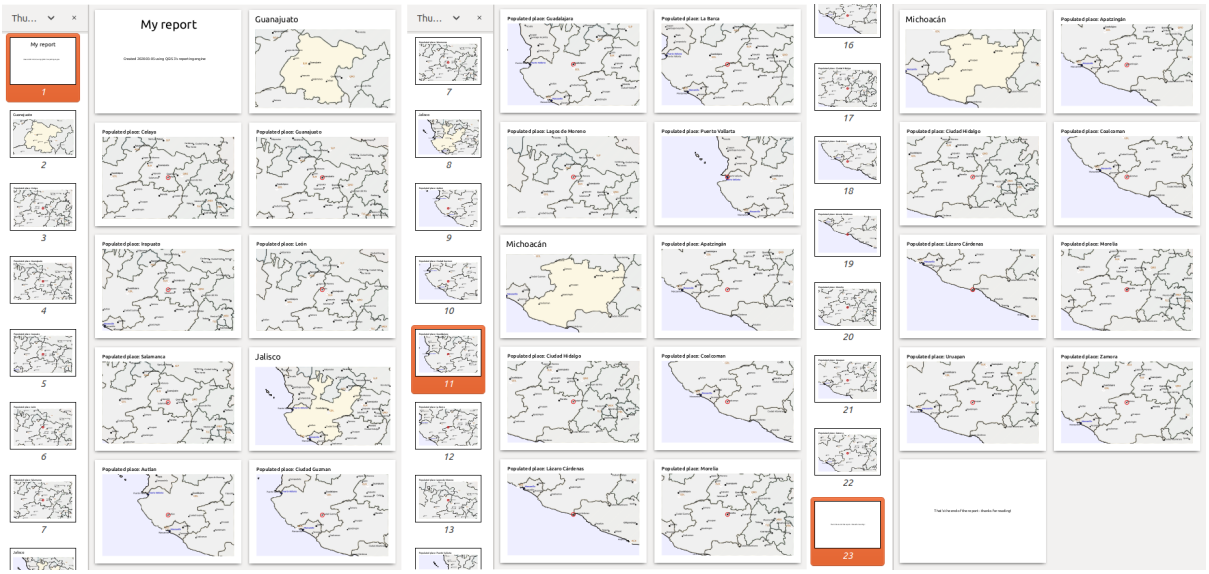
So more or less an atlas, but with a header and footer page.

Let us make things more interesting by adding a subsection to our state group. We do this by first selecting the *Admin Level 1* field group in the organizer, then hitting the Add Field button and adding a new *Field Group Section*:



When iterating over the features of a *Field Group Section*, the features will be filtered to match the defining field

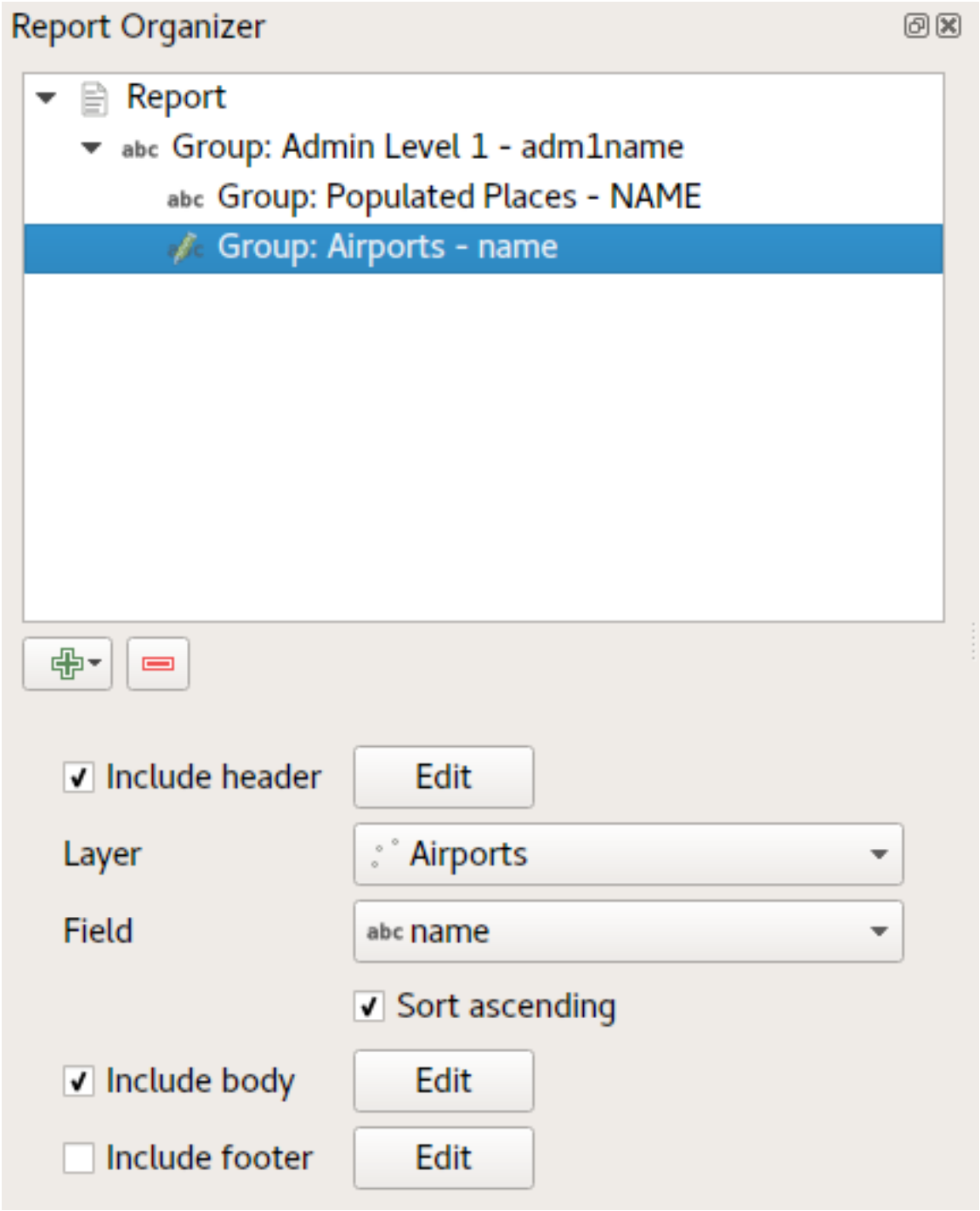
of its parent group (`adm1name` in this case). Here, the subsection we added will iterate over a *Populated Places* layer, including a body section for each place encountered. The magic here is that the *Populated Places* layer has an attribute with the same name as the defining field in the parent layer, *adm1name*, tagging each place with the state it is contained within (if you're lucky your data will already be structured like this – if not, run the *Join Attributes by Location* Processing algorithm and create your own field). When we export this report, QGIS will grab the first state from the *Admin Level 1* layer, and then iterate over all the *Populated Places* with a matching *adm1name* value. Here's what we get:



Here we created a basic body for the Populated Places group, including a map of the place and a table of some place attributes. So our report is now a report header, a page for the first state, followed by a page for every populated place within that state, then the rest of the states with their populated places, and finally the report footer. If we were to add a header for the Populated Places group, it would be included just before listing the populated places for each state, as shown in the illustration below.

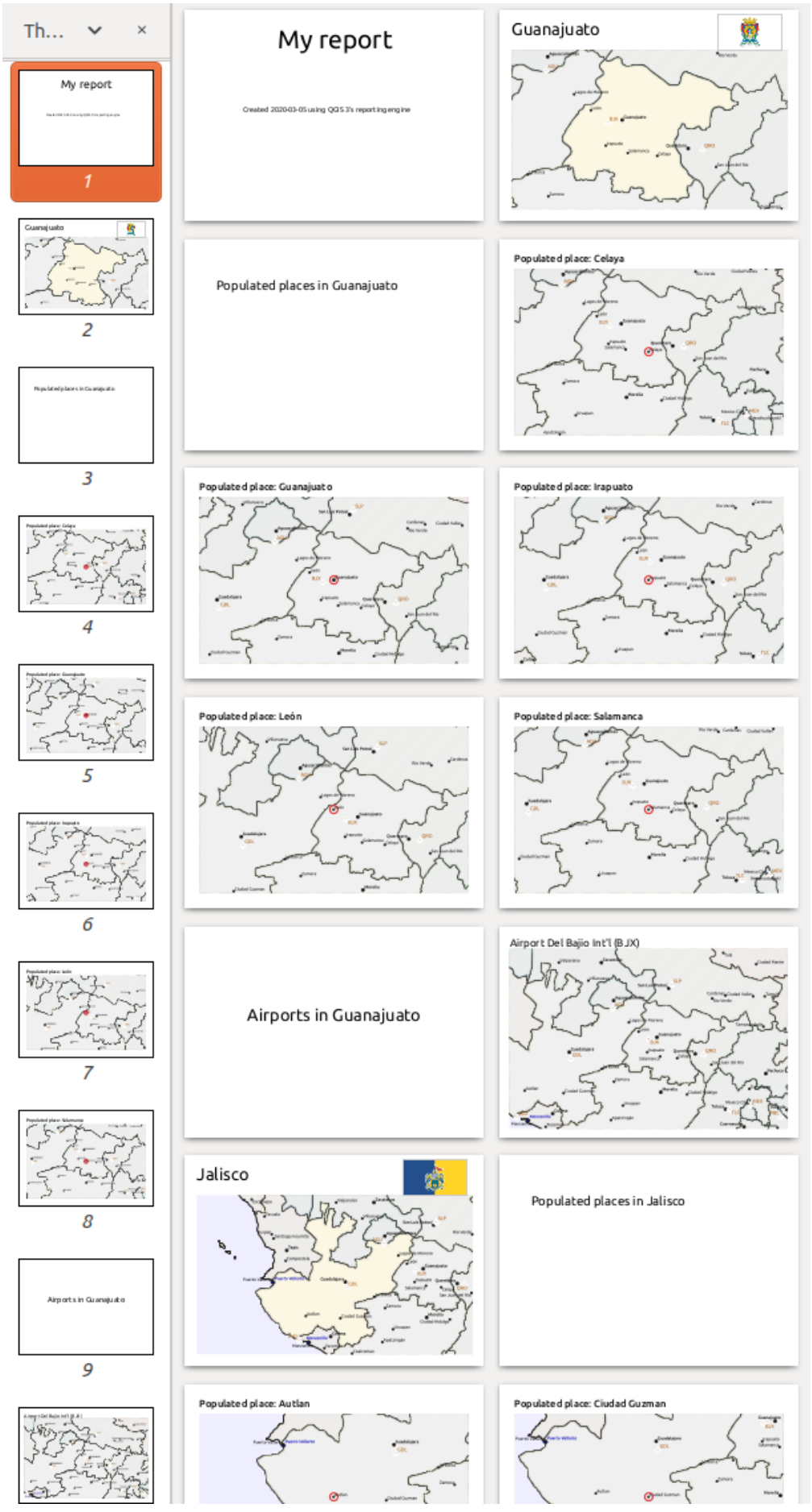
Similarly, a footer for the Populated Places group would be inserted after the final place for each state is included.

In addition to nested subsections, subsections in a report can also be included consecutively. If we add a second subsection to the *Admin Level 1 group* for *Airports*, then (if the *Airports* layer has an attribute *adm1name* that can link it to the parent group) our report will first list ALL the populated places for each state, followed by all the airports within that state, before proceeding to the next state.



The key point here is that our *Airports group* is a subsection of the *Admin Level 1 group* – not the *Populated Places group*.

In this case our report would be structured like this (note that state flags have also been included - the procedure for adding feature specific pictures in this way is described below):


Th...
My report
1
2
3
4
5
6
7
8
9
My report
Created 2020-03-05 using QGIS 3's reporting engine
Guanajuato
Populated places in Guanajuato
Populated place: Celaya
Populated place: Guanajuato
Populated place: Irapuato
Populated place: León
Populated place: Salamanca
Airports in Guanajuato
Airport Del Bajio Int'l (BJX)
Jalisco
Populated places in Jalisco
Populated place: Autlan
Populated place: Ciudad Guzman

Including pictures in a report

Pictures can be quite useful in reports, and QGIS allows pictures in both the static and dynamic parts of a report. Pictures are added in the same way as for standard print layouts, and for the static report parts (and static pictures in dynamic parts) there is not more to it.

But if you want illustrations that are tailored to the report features, your layer must have an attribute that can be used to define the picture to include.

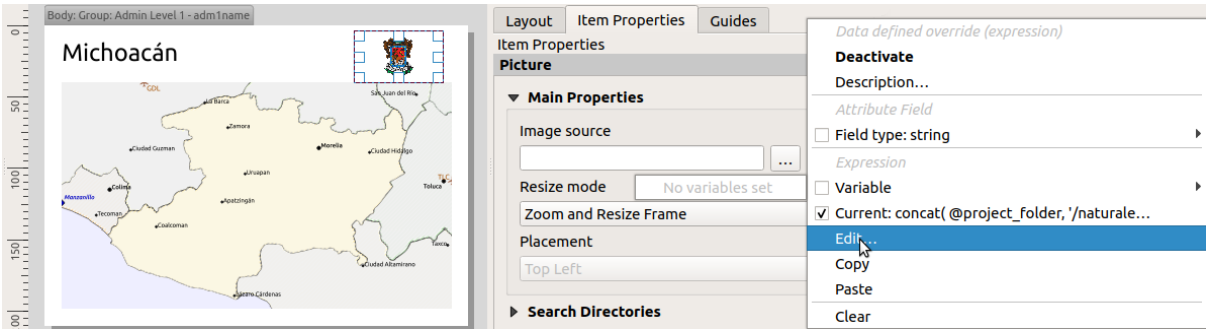
QGIS depends on absolute file names for images in reports.

For dynamic pictures, you first add a picture to the body part of the group, as usual. In the *Item properties* of the picture, you set the *Image Source* using the Data defined override button, and either select an attribute that contains the absolute path of the images or *Edit...* (to enter an expression that generates the absolute image path).

Below is an example expression that uses string concatenation to specify the absolute path to the pictures, using the directory where the project file is located (`@project_path`) and an attribute (`adm1name`) from which the file name is generated (in this case by transforming the string in the `adm1name` attribute to uppercase, and appending ‚_flag.png‘):

```
concat(@project_folder, '/naturalearth/pictures/' ,
       upper("adm1name"), '_flag.png')
```

This means that the pictures are located in the `naturalearth/pictures` subdirectory of the project file directory.



Highlighting the current report feature in a map

In the above report, the report features are emphasized in the maps using highlighting (state) and circles (populated places). To emphasize the report features in the maps (apart from placing them at the centre of the maps), you must data define the style using a comparison between its `@id` and the `@atlas_featureid`, as for atlases.

For instance, if you would like to use a thicker line / border for the report feature than the other features you can data define the line width:

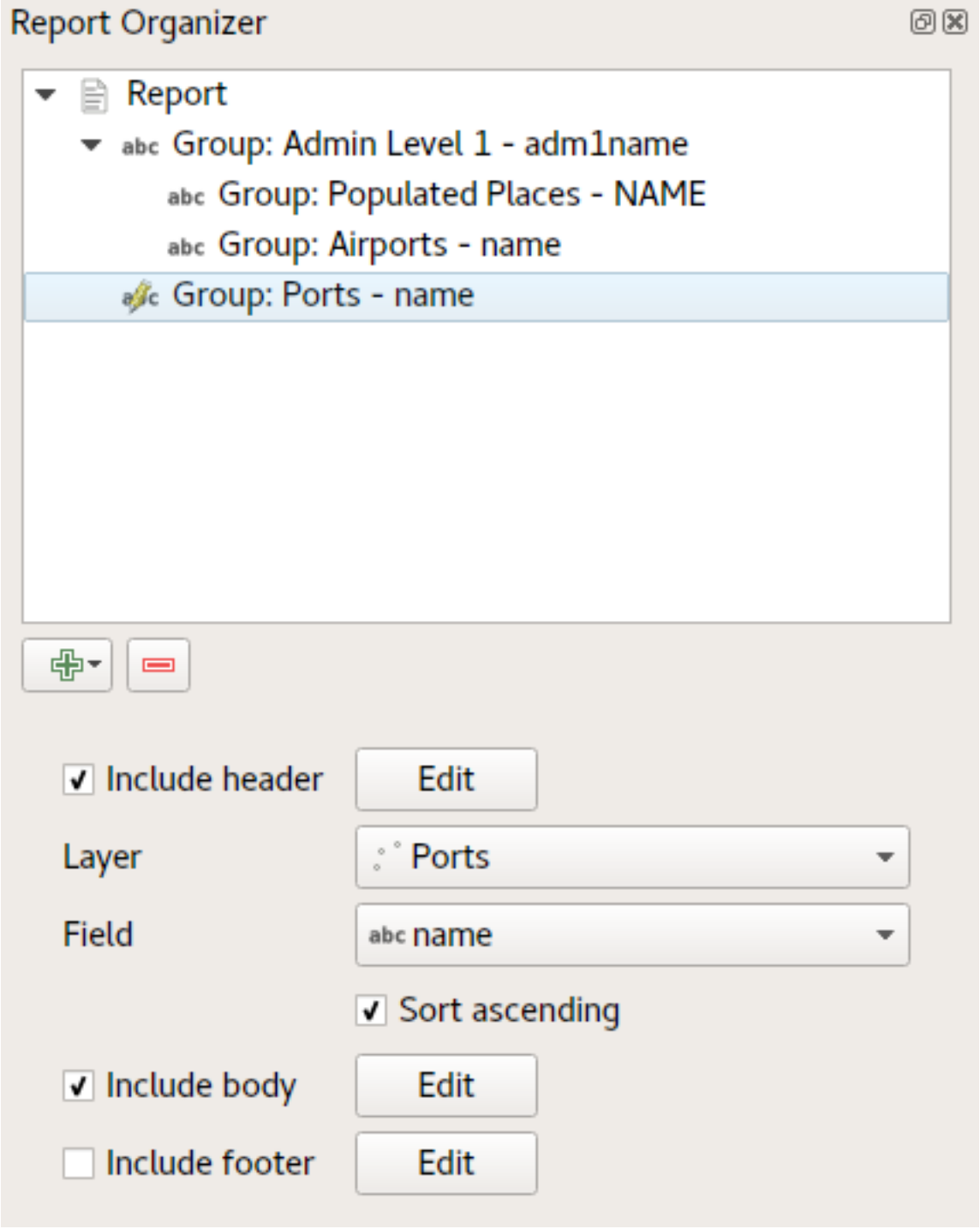
```
if($id=@atlas_featureid, 2.0, 0.1)
```

The report feature will get a 2 units wide polygon outline, while all other features will get a 0.1 units wide line. It is also possible to data define the colour (non-transparent dark magenta for the report feature and semi-transparent light gray for the other features):

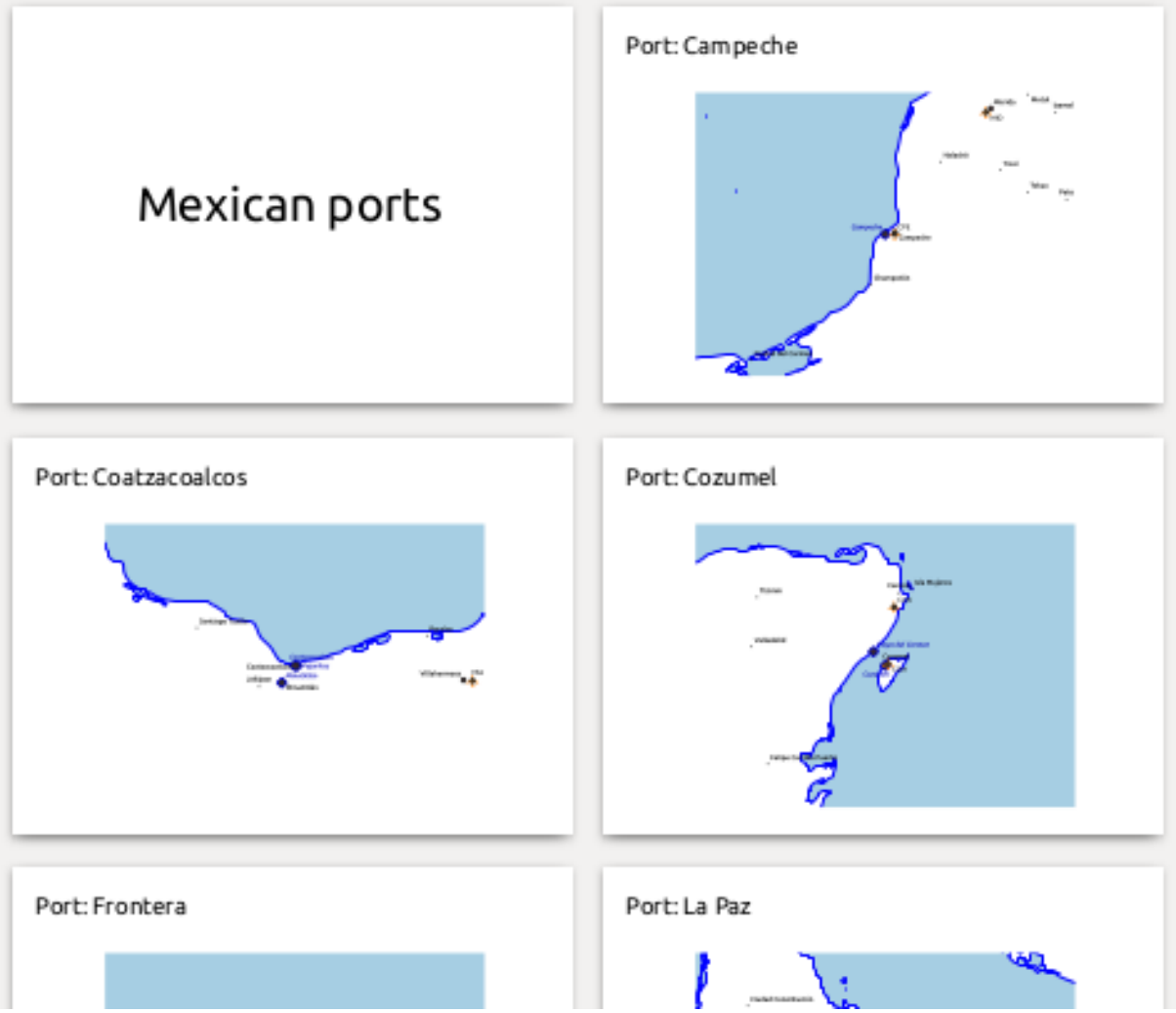
```
if($id=@atlas_featureid, '#FF880088', '#88CCCCCC')
```

More level 1 groups

Combining nested and consecutive sections, together with section headers and footers allows for tons of flexibility. For instance, in the below report we add another field group as a child of the main report for the *Ports* layer. Now, after listing the states together with their populated places and airports, we'll get a summary list of all the ports in the region:



This results in the last part of our report exporting as:



18.4.4 Eksporto nustatymai

When you export a report (*Report* ► *Export Report as Images… / SVG… / PDF…*), you will be asked for a file name, and then you get the opportunity to tune the export settings to get the most appropriate output.

As you see, reports in QGIS are extremely powerful and flexible!

Pastaba

The current information was adapted from a North Road blog, Exploring Reports in QGIS 3.0 - the Ultimate Guide!

SKYRIUS 19

Darbas su OGC / ISO protokolais

Atviras Geoerdvinis Konsorciumas (OGC) yra tarptautinė organizacija su nariais iš daugiau nei 300 viso pasaulio komercinių, valstybinių, nepelno ir tyrimo organizacijų. Jo nariai vysto ir įgyvendina erdvinio turinio ir paslaugų standartus, GIS duomenų apdorojimą ir apsikeitimą.

Describing a basic data model for geographic features, an increasing number of specifications are developed by OGC to serve specific needs for interoperable location and geospatial technology, including GIS. Further information can be found at https://www.ogc.org/.

Important OGC specifications supported by QGIS are:

- **WMS** — Web Map Service (*WMS/WMTS Client*)
- **WMTS** — Web Map Tile Service (*WMS/WMTS Client*)
- **WFS** — Web Feature Service (*WFS and WFS-T Client*)
- **WFS-T** — Web Feature Service - Transactional (*WFS and WFS-T Client*)
- **WCS** — Web Coverage Service (*WCS Client*)
- **WPS** — Web Processing Service
- **CSW** — Catalog Service for the Web
- **SFS** — Simple Features for SQL (*PostgreSQL Database*)
- **GML** — Geography Markup Language
- **SensorThings API** — SensorThings API (*SensorThings*)

OGC services are increasingly being used to exchange geospatial data between different GIS implementations and data stores. QGIS can deal with the above specifications as a client, being **SFS** (through support of the PostgreSQL data provider, see section *PostgreSQL Database*).

You can also share your maps and data through the WMS, WMTS, WFS, WFS-T and WCS protocols using a webserver with QGIS Server, UMN MapServer or GeoServer installed.

19.1 WMS/WMTS Client

19.1.1 Overview of WMS Support

QGIS currently can act as a WMS client that understands WMS 1.1, 1.1.1 and 1.3 servers. In particular, it has been tested against publicly accessible servers such as DEMIS.

A WMS server acts upon requests by the client (e.g., QGIS) for a raster map with a given extent, set of layers, symbolization style, and transparency. The WMS server then consults its local data sources, rasterizes the map, and sends it back to the client in a raster format. For QGIS, this format would typically be JPEG or PNG.

WMS is generically a REST (Representational State Transfer) service rather than a full-blown Web service. As such, you can actually take the URLs generated by QGIS and use them in a web browser to retrieve the same images that QGIS uses internally. This can be useful for troubleshooting, as there are several brands of WMS server on the market and they all have their own interpretation of the WMS standard.

WMS layers can be added quite simply, as long as you know the URL to access the WMS server, you have a serviceable connection to that server, and the server understands HTTP as the data transport mechanism.

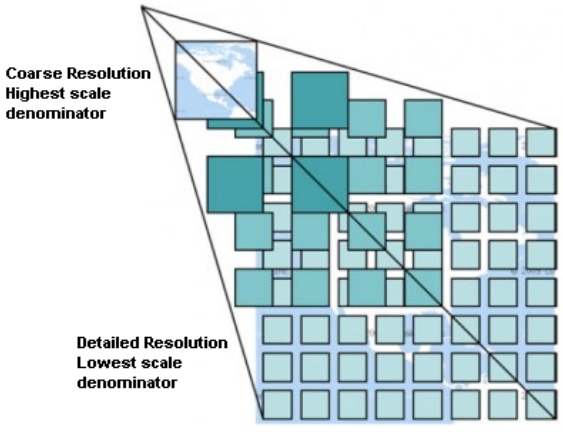
Additionally, QGIS will cache your WMS responses (i.e. images) for 24h as long as the GetCapabilities request is not triggered. The GetCapabilities request is triggered every time the *Connect* button in the *WMS/WMTS* dialog is used to retrieve the WMS server capabilities. This is an automatic feature meant to optimize project loading time. If a project is saved with a WMS layer, the corresponding WMS tiles will be loaded from the cache the next time the project is opened as long as they are not older than 24h.

19.1.2 Overview of WMTS Support

QGIS can also act as a WMTS client. WMTS is an OGC standard for distributing tile sets of geospatial data. This is a faster and more efficient way of distributing data than WMS because with WMTS, the tile sets are pre-generated, and the client only requests the transmission of the tiles, not their production. A WMS request typically involves both the generation and transmission of the data. A well-known example of a non-OGC standard for viewing tiled geospatial data is Google Maps.

In order to display the data at a variety of scales close to what the user might want, the WMTS tile sets are produced at several different scale levels and are made available for the GIS client to request them.

This diagram illustrates the concept of tile sets:



19.1 Fig. : Concept of WMTS tile sets

The two types of WMTS interfaces that QGIS supports are via Key-Value-Pairs (KVP) and RESTful. These two interfaces are different, and you need to specify them to QGIS differently.

1. In order to access a **WMTS KVP** service, a QGIS user must open the WMS/WMTS interface and add the following string to the URL of the WMTS tile service:

```
"?SERVICE=WMTS&REQUEST=GetCapabilities"
```

An example of this type of address is:

```
https://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?service=WMTS&
↪request=GetCapabilities
```

For testing the topo2 layer in this WMTS works nicely. Adding this string indicates that a WMTS web service is to be used instead of a WMS service.

2. The **RESTful WMTS** service takes a different form, a straightforward URL. The format recommended by the OGC is:

```
{WMTSBaseURL}/1.0.0/WMTSCapabilities.xml
```

This format helps you to recognize that it is a RESTful address. A RESTful WMTS is accessed in QGIS by simply adding its address in the WMS setup in the URL field of the form. An example of this type of address for the case of an Austrian basemap is:

```
https://maps.wien.gv.at/basemap/1.0.0/WMTSCapabilities.xml
```

Pastaba

You can still find some old services called WMS-C. These services are quite similar to WMTS (i.e., same purpose but working a little bit differently). You can manage them the same as you do WMTS services. Just add `?tiled=true` at the end of the url. See https://wiki.osgeo.org/wiki/Tile_Map_Service_Specification for more information about this specification.

When you read WMTS, you can often think WMS-C also.

19.1.3 Selecting WMS/WMTS Servers

The first time you use the WMS/WMTS feature in QGIS, there are no servers defined.

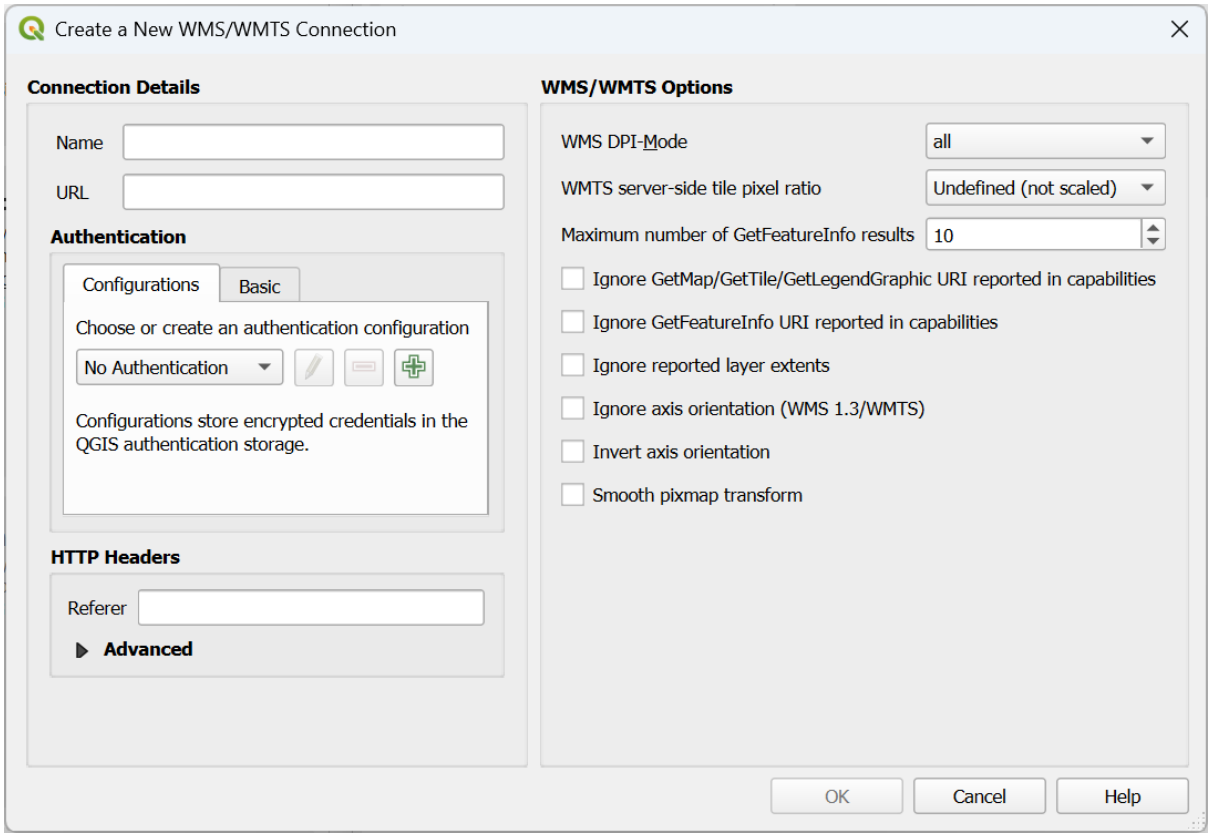
You then need to create connections to the server you are targeting:

1. Go to the *WMS/WMTS* tab of the *Data Source Manager* dialog, either by:
   - clicking the Open Data Source Manager button (or pressing `Ctrl+L`) and enabling the tab
   - clicking the Add WMS layer button on the *Manage Layers* toolbar
   - or selecting *Layer* ► *Add Layer* ► *Add WMS/WMTS Layer…* menu
2. Press *New* from the *Layers* tab. The *Create a New WMS/WMTS Connection…* dialog appears.

Patarimas

Right-click the *WMS/WMTS* entry from within the *Browser panel* and select *New Connection…* also opens the *Create a New WMS/WMTS Connection…* dialog.

3. Then enter the parameters to connect to your desired WMS server, as listed below:



19.2 Fig. : Creating a connection to a WMS server

- *Name*: A name for the connection. This name will be used in the Server Connections drop-down box so that you can distinguish it from other WMS servers.
- *URL*: URL of the server providing the data. This must be a resolvable host name – the same format as you would use to open a telnet connection or ping a host, i.e. the base URL only. For example, you shouldn't have fragments such as `request=GetCapabilities` or `version=1.0.0` in your URL.
- *Authentication* (optional): using a *stored configuration* or a basic authentication with *Username* and *Password*.

> **Įspėjimas**
>
> Entering **username** and **password** in the *Authentication* tab will keep unprotected credentials in the connection configuration. Those **credentials will be visible** if, for instance, you shared the project file with someone. Therefore, it's advisable to save your credentials in an *Authentication configuration* instead (*Configurations* tab). See *Autentifikacijos sistema* for more details.

- HTTP *Referer*
- *WMS DPI-Mode*: Available options are **all**, **off**, **QGIS**, **UMN** and **GeoServer**
- *WMTS server-side tile pixel ratio*: When rendering WMTS layers, allows to scale up or down the tiles based on the device screen DPI. Available options are *Undefined (not scaled)*, *Standard (96 DPI)* and *High (192 DPI)*.
- *Maximum number of GetFeatureInfo results*: specifies a default value for the maximum number of results returned per layer by a GetFeatureInfo request using this connection (see FEATURE_COUNT parameter). Default value is `10`. Set to `0` to use server default value (usually `1`): no FEATURE_COUNT parameter will be added to the request.

- ☐ *Ignore GetMap/GetTile/GetLegendGraphic URI reported in capabilities*: if checked, use given URI from the *URL* field above.
- ☐ *Ignore GetFeatureInfo URI reported in capabilities*: if checked, use given URI from the *URL* field above.
- ☐ *Ignore reported layer extents*: because the extent reported by raster layers may be smaller than the actual area which can be rendered (notably for WMS servers with symbology which takes more space than the data extent), check this option to avoid cropping raster layers to their reported extents, resulting in truncated symbols on the borders of these layers.
- ☐ *Ignore axis orientation (WMS 1.3/WMTS)*
- ☐ *Invert axis orientation*
- ☐ *Smooth pixmap transformation*

4. Spauskite *Gerai*

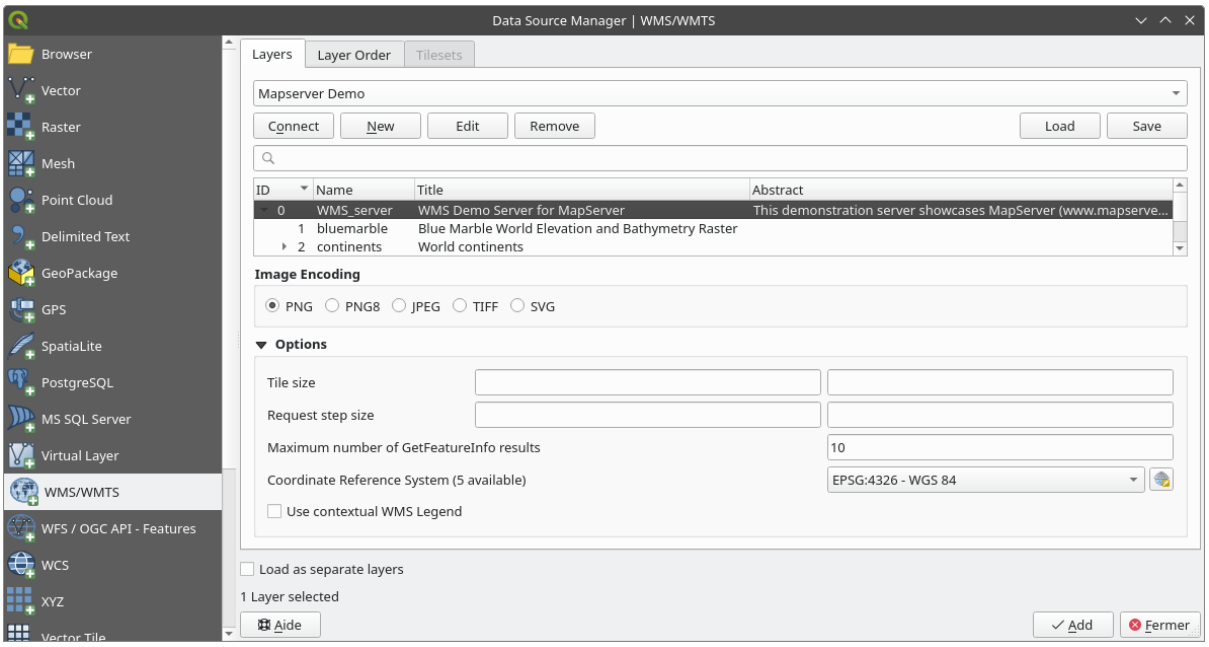
Once the new WMS/WMTS server connection has been created, it will be preserved for future QGIS sessions. Note that it is also possible to *Load* the connection parameters from a `.XML` file or *Save* them to a `.XML` file.

If you need to set up a proxy server to be able to receive WMS services from the internet, you can add your proxy server in the options. Choose *Settings* ► *Options* and click on the *Network* tab. There, you can add your proxy settings and enable them by setting ☑ *Use proxy for web access*. Make sure that you select the correct proxy type from the *Proxy type* [...▼] drop-down menu.

19.1.4 Loading WMS/WMTS Layers

Once you have successfully filled in your parameters, you can use the *Connect* button to retrieve the capabilities of the selected server. This includes the image encoding, layers, layer styles and projections. Since this is a network operation, the speed of the response depends on the quality of your network connection to the WMS server. While downloading data from the WMS server, the download progress is visualized in the lower left corner of the main QGIS dialog.

Your screen should now look a bit like Fig. 19.3, which shows the response provided by a WMS server.



19.3 Fig. : Dialog for adding a WMS server, with filter on available layers

The upper part of the *Layers* tab of the dialog shows a tree structure that can include layer groups embedding layers with their associated image style(s) served by the server. Each item can be identified by:

- an *ID*
- *Pavadinimą*
- a *Title*
- and an *Abstract*.

The list can be filtered using the widget in the top right corner.

Image Encoding

The *Image encoding* section lists the formats that are supported by both the client and server. Choose one depending on your image accuracy requirements.

Patarimas

Image Encoding

You will typically find that a WMS server offers you the choice of JPEG or PNG image encoding. JPEG is a lossy compression format, whereas PNG faithfully reproduces the raw raster data.

Use JPEG if you expect the WMS data to be photographic in nature and/or you don't mind some loss in picture quality. This trade-off typically reduces by five times the data transfer requirement compared with PNG.

Use PNG if you want precise representations of the original data and you don't mind the increased data transfer requirements.

Options

The Options area of the dialog provides means to configure the WMS requests. You can define:

- *Tile size* if you want to set tile sizes (e.g., 256x256) to split up the WMS request into multiple requests.
- *Request step size*: if you want to reduce the effect of cut labels at tile borders, increasing the step size creates larger requests, fewer tiles and fewer borders. The default value is 2000.
- The *Maximum number of GetFeatureInfo results*: specifies the maximum number of results returned by a GetFeatureInfo request, for the layer(s) being loaded. Allows to override the *connection's default value* for specific layers.
- Each WMS layer can be presented in multiple CRSs, depending on the capability of the WMS server. If you select a WMS from the list, a field with the default projection provided by the web server appears. Press the Select CRS widget to replace the default projection of the WMS with another CRS supported by the WMS server.

  A dialog similar to the one shown in Fig. 6.4 will appear. The main difference with the WMS version of the dialog is that only those CRSs supported by the WMS server will be shown.
- Finally you can activate *Use contextual WMS Legend* if the WMS Server supports this feature. Then only the relevant legend for your current map view extent will be shown and thus will not include legend items for items you can't see in the current map.

You can select several layers at once, but only one image style per layer. When several layers are selected, they will be combined at the WMS server and transmitted to QGIS in one go, as a single layer. The default name is a slash (/) separated list of their original title. You can however opt to *Load as separate layers*, in which case each layer is added to the map canvas using its default name from the queried service.

Layer Order

The *Layer Order* tab lists the selected layers available from the current connected WMS server.

WMS layers rendered by a server are overlaid in the order listed in the *Layers* tab, from top to bottom of the list. If you want to change the overlay order, you can use the *Up* and *Down* buttons of the *Layer Order* tab.

Transparency

The *Global transparency* setting from the *Layer Properties* is hard coded to be always on, where available.

19.1.5 Kaladėlių rinkiniai

When using WMTS (Cached WMS) services you are able to browse through the *Tilesets* tab given by the server. Additional information like tile size, formats and supported CRS are listed in this table.

```
# example of WMTS service

https://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?service=WMTS&
↪request=GetCapabilities
```

Selecting a layer to load, it is also possible to apply an *Interpretation method*, converting it into a raster layer of single band float type, ready for styling using QGIS usual *raster renderers*.

In combination with this feature, you can use the tile scale slider by selecting *View* ► *Panels* (or *Settings* ► *Panels*), then choosing *Tile Scale Panel*. This gives you the available scales from the tile server with a nice slider docked in.

19.1.6 Using the Identify Tool

Once you have added a WMS server, and if any layer from a WMS server is queryable, you can then use the Identify tool to select a pixel on the map canvas. A query is made to the WMS server for each selection made. The results of the query are returned in plain text. The formatting of this text is dependent on the particular WMS server used.

Format selection

If multiple output formats are supported by the server, a combo box with supported formats is automatically added to the identify results dialog and the selected format may be stored in the project for the layer.

GML format support

The Identify tool supports WMS server response (GetFeatureInfo) in GML format (it is called Feature in the QGIS GUI in this context). If „Feature“ format is supported by the server and selected, results of the Identify tool are vector features, as from a regular vector layer. When a single feature is selected in the tree, it is highlighted in the map and it can be copied to the clipboard and pasted to another vector layer. See the example setup of the UMN Mapserver below to support GetFeatureInfo in GML format.

```
# in layer METADATA add which fields should be included and define geometry␣
↪(example):

"gml_include_items"   "all"
"ows_geometries"      "mygeom"
"ows_mygeom_type"     "polygon"

# Then there are two possibilities/formats available, see a) and b):

# a) basic (output is generated by Mapserver and does not contain XSD)
# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "application/vnd.ogc.gml,text/html"

# b) using OGR (output is generated by OGR, it is sent as multipart and contains␣
↪XSD)
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
# in MAP define OUTPUTFORMAT (example):
OUTPUTFORMAT
    NAME "OGRGML"
    MIMETYPE "ogr/gml"
    DRIVER "OGR/GML"
    FORMATOPTION "FORM=multipart"
END

# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "OGRGML,text/html"
```

19.1.7 Viewing Properties

Once you have added a WMS server, you can view its properties by right-clicking on it in the legend and selecting *Properties*. The WMS/WMTS layer properties is much like the raster layer properties so you will find detailed description at *Rastro savybių dialogas*. However, there are some differences, which will be explained below.

Information properties

Metadata Tab

The tab *Metadata* displays a wealth of information about the WMS server, generally collected from the capabilities statement returned from that server. Many definitions can be gleaned by reading the WMS standards (see OPEN-GEOSPATIAL-CONSORTIUM in *Literature and Web References*), but here are a few handy definitions:

- **Server Properties**
  - **WMS Version** — The WMS version supported by the server.
  - **Image Formats** — The list of MIME-types the server can respond with when drawing the map. QGIS supports whatever formats the underlying Qt libraries were built with, which is typically at least `image/png` and `image/jpeg`.
  - **Identity Formats** — The list of MIME-types the server can respond with when you use the Identify tool. Currently, QGIS supports the `text-plain` type.
- **Layer Properties**
  - **Selected** — Whether or not this layer was selected when its server was added to this project.
  - **Visible** — Whether or not this layer is selected as visible in the legend (not yet used in this version of QGIS).
  - **Can Identify** — Whether or not this layer will return any results when the Identify tool is used on it.
  - **Can be Transparent** — Whether or not this layer can be rendered with transparency. This version of QGIS will always use transparency if this is `Yes` and the image encoding supports transparency.
  - **Can Zoom In** — Whether or not this layer can be zoomed in by the server. This version of QGIS assumes all WMS layers have this set to `Yes`. Deficient layers may be rendered strangely.
  - **Cascade Count** — WMS servers can act as a proxy to other WMS servers to get the raster data for a layer. This entry shows how many times the request for this layer is forwarded to peer WMS servers for a result.
  - **Fixed Width, Fixed Height** — Whether or not this layer has fixed source pixel dimensions. This version of QGIS assumes all WMS layers have this set to nothing. Deficient layers may be rendered strangely.
  - **WGS 84 Bounding Box** — The bounding box of the layer, in WGS 84 coordinates. Some WMS servers do not set this correctly (e.g., UTM coordinates are used instead). If this is the case, then the initial view of this layer may be rendered with a very ‚zoomed-out' appearance by QGIS. The WMS webmaster should be informed of this error, which they may know as the WMS XML elements `LatLonBoundingBox`, `EX_GeographicBoundingBox` or the CRS:84 `BoundingBox`.

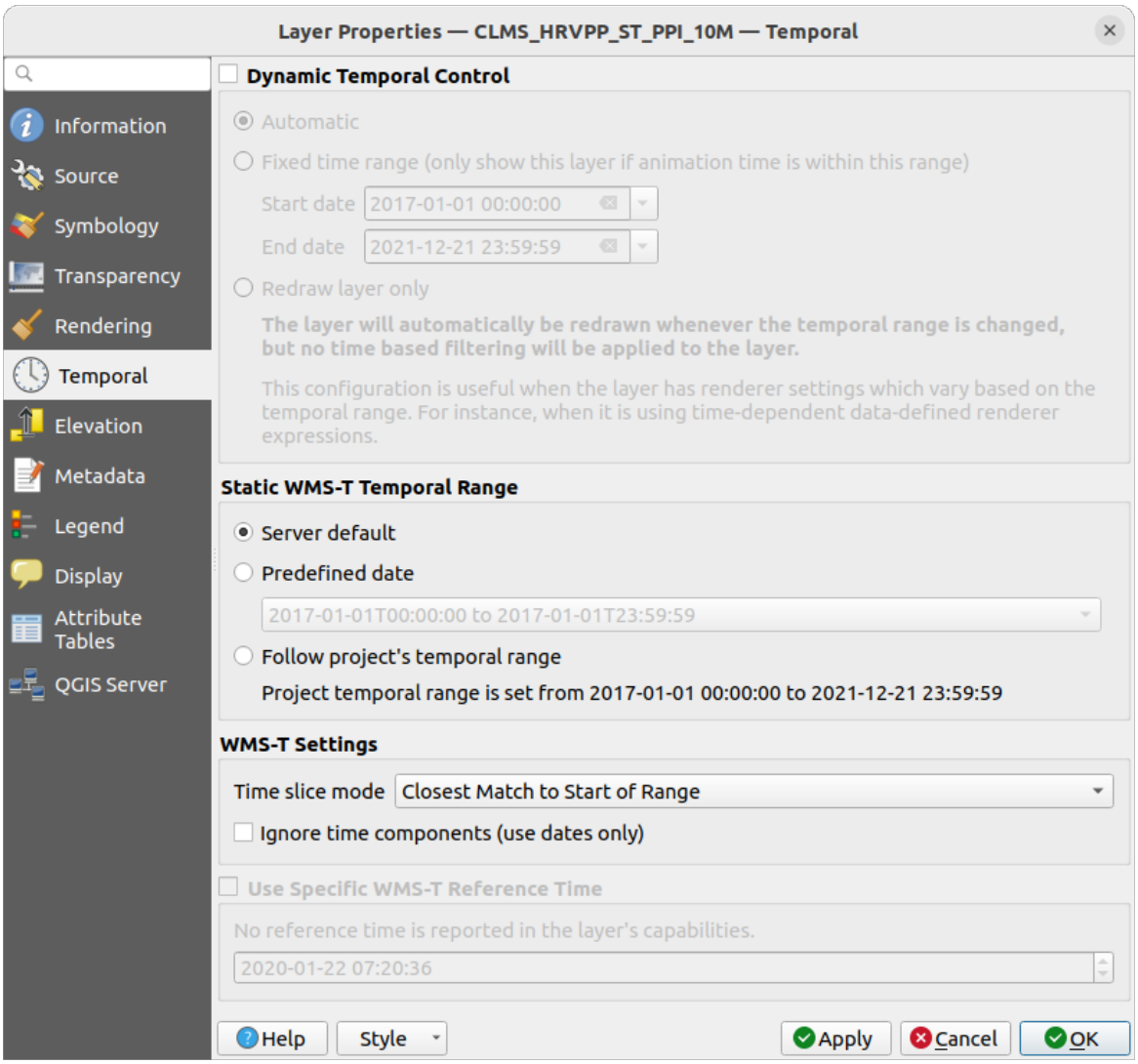
- **Available in CRS —** The projections that this layer can be rendered in by the WMS server. These are listed in the WMS-native format.
- **Available in style —** The image styles that this layer can be rendered in by the WMS server.

Temporal properties

Raster *temporal properties* (namely *Dynamic Temporal Control*) can be set for WMS and WMTS layers. By default, when a time-dimension enabled WMS or WMTS layer is added to the project, it is indicated in the *Layers* panel with the Temporal Layer icon next to it. Its *Temporal* properties default to the *Automatic* temporal mode, meaning that the layer will follow the temporal controller's current time range by default.

You can then opt to show a specific static time value for the layer by unchecking *Dynamic Temporal Control* and picking an option under *Static WMS-T Temporal Range*:

- *Server default*
- *Predefined date* with a server exposing data for non-contiguous temporal ranges or *Predefined range* with a server exposing a range of available dates. A *Start date* and *End date* are necessary in the latter case. Their expected formatting can be deduced from the reference time option (see below). depending on whether the provider has data for contiguous period or not
- *Follow project's temporal range* as defined in the project's properties dialog



19.4 Fig. : Temporal properties of a WMTS layer

Whatever temporal data control is in use, there are some *WMS-T Settings* to help display the correct temporal data:

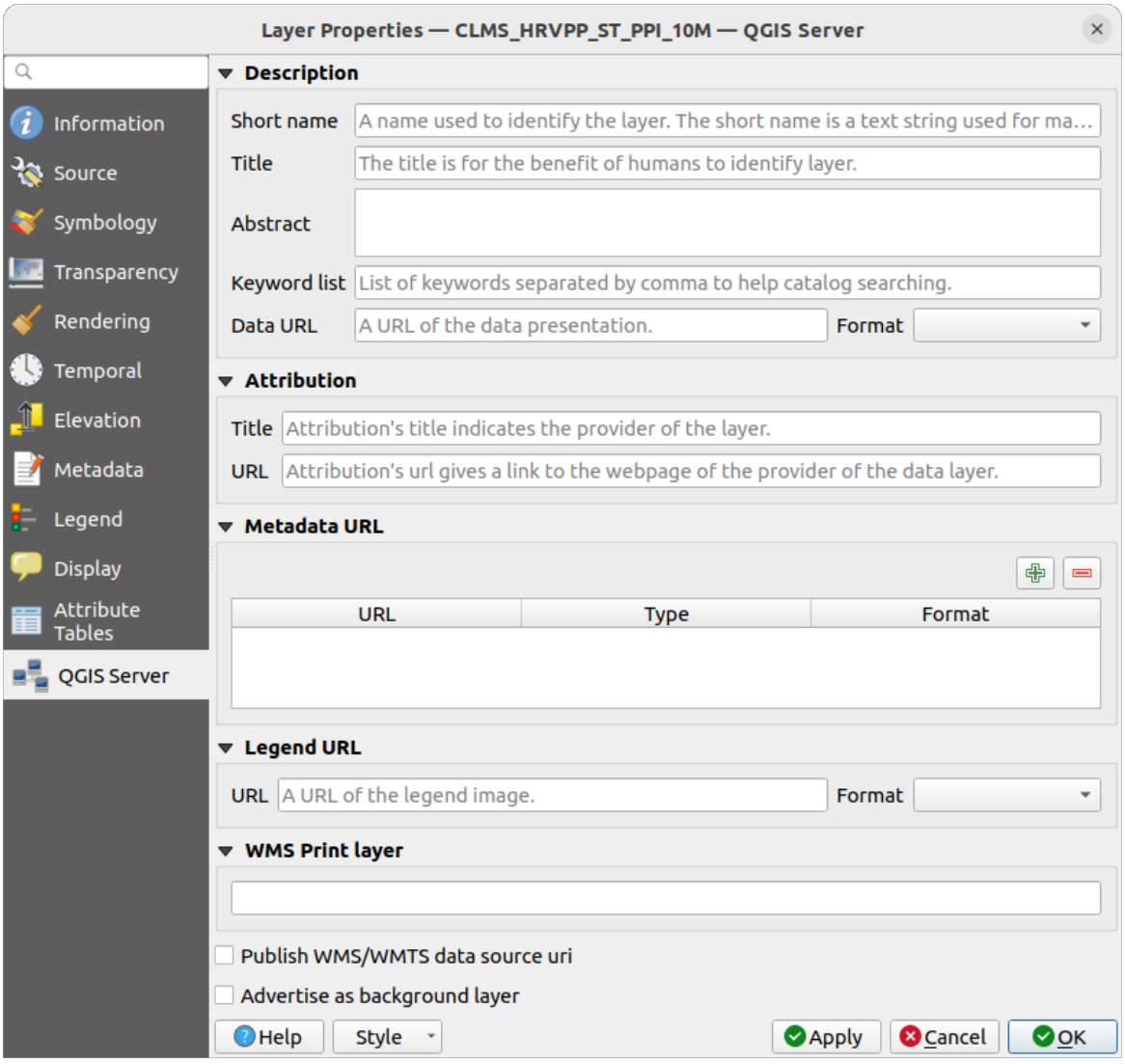
- *Time slice mode* which can be:
  - *Use whole temporal range*
  - *Match to start of range*
  - *Match to end of range*
  - *Closest match to start of range*
  - *Closest match to end of range*
- *Ignore time components (use dates only)*: If checked, the time component of temporal queries will be discarded and only the date component will be used in server requests.

You can also *Use Specific WMS-T Reference Time* picked from times reported in the layer's capabilities. Convenient for servers which expose a non-contiguous set of date time instances (instead of a range of dates).

QGIS Server properties

In addition to the raster layer *properties*, publishing a WMS/WMTS layer with QGIS Server will display following options:

- *WMS Print layer*: Allows to set alternative WMS layer that should be used for printing (in the GetProjectSettings reply). Convenient for WMTS layers that are generally not suitable for printing.
- ☑ *Publish WMS/WMTS data source uri*: Allows a web client to fetch the WMS/WMTS data directly
- ☑ *Advertise as background layer*



19.5 Fig. : QGIS Server properties of a WMS/WMTS layer

19.1.8 Show WMS legend graphic in table of contents and layout

The QGIS WMS data provider is able to display a legend graphic in the table of contents' layer list and in the print layout. The WMS legend will be shown only if the WMS server has GetLegendGraphic capability and the layer has getCapability url specified, so you additionally have to select a styling for the layer.

If a legendGraphic is available, it is shown below the layer. It is little and you have to click on it to open it in real dimension (due to QgsLegendInterface architectural limitation). Clicking on the layer's legend will open a frame with the legend at full resolution.

In the print layout, the legend will be integrated at it's original (downloaded) dimension. Resolution of the legend graphic can be set in the item properties under *Legend* ► *WMS LegendGraphic* to match your printing requirements.

The legend will display contextual information based on your current scale. The WMS legend will be shown only if the WMS server has GetLegendGraphic capability and the layer has getCapability url specified, so you have to select a styling.

19.2 WCS Client

A Web Coverage Service (WCS) provides access to raster data in forms that are useful for client-side rendering, as input into scientific models, and for other clients. The WCS may be compared to the WFS and the WMS. As WMS and WFS service instances, a WCS allows clients to choose portions of a server's information holdings based on spatial constraints and other query criteria.

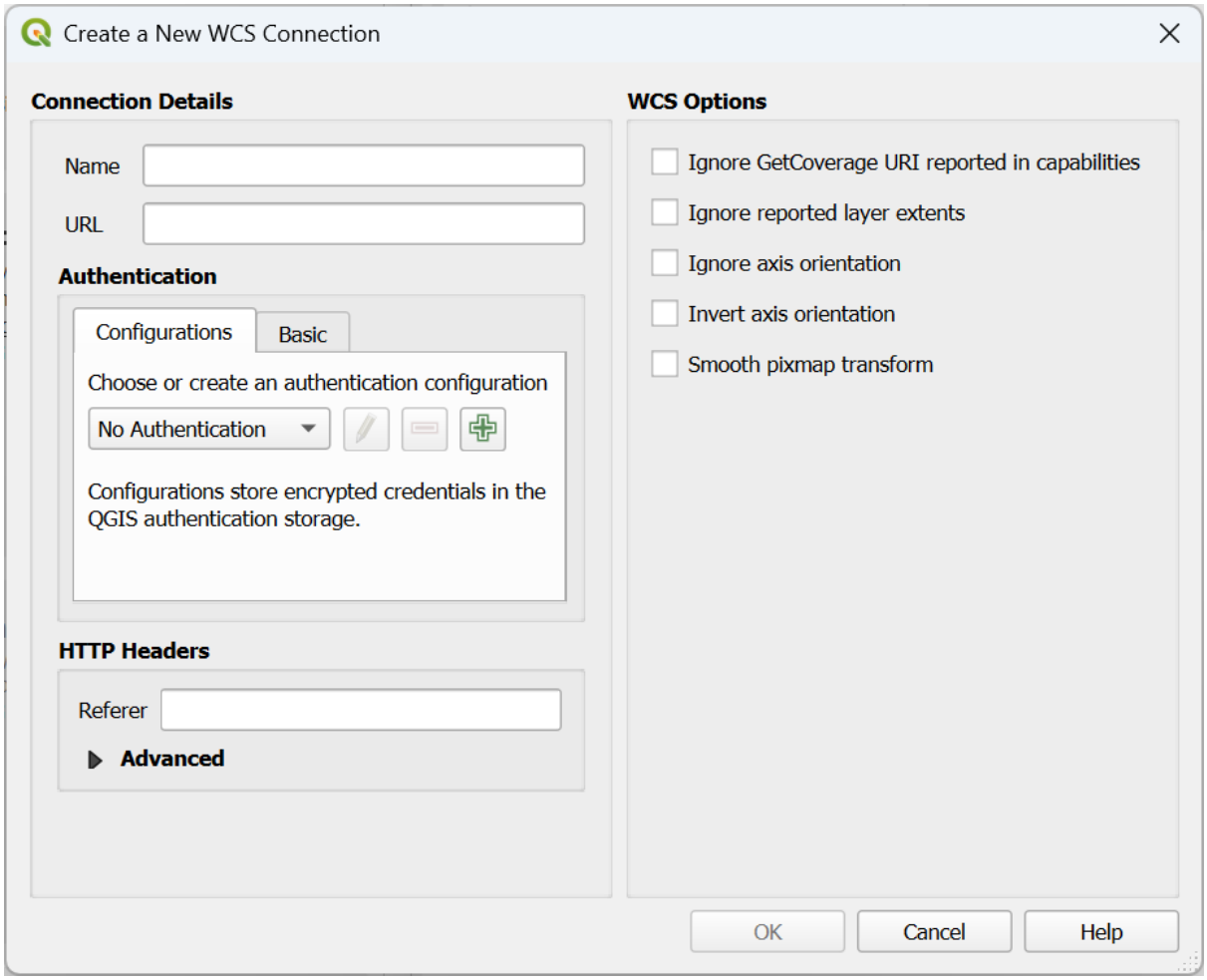
QGIS has a native WCS provider and supports both version 1.0 and 1.1 (which are significantly different), but currently it prefers 1.0, because 1.1 has many issues (i.e., each server implements it in a different way with various particularities).

The native WCS provider handles all network requests and uses all standard QGIS network settings (especially proxy). It is also possible to select cache mode (‚always cache', ‚prefer cache', ‚prefer network', ‚always network'), and the provider also supports selection of time position, if temporal domain is offered by the server.

Loading a WCS Layer

To be able to load a WCS Layer, first create a connection to the WCS server:

1. Open the *Data Source Manager* dialog by pressing the Open Data Source Manager button
2. Enable the *WCS* tab
3. Click on *New…* to open the *Create a New WCS Connection* dialog



19.6 Fig. : Creating a connection to a WCS server

- *Name*: A name for the connection. This name will be used in the Server Connections drop-down box so that you can distinguish it from other WCS servers.
- *URL*: URL of the server providing the data. This must be a resolvable host name – the same format as you would use to open a telnet connection or ping a host, i.e. the base URL only. For example, you shouldn't have fragments such as `request=GetCapabilities` or `version=1.0.0` in your URL.
- *Authentication* (optional): using a *stored configuration* or a basic authentication with *Username* and *Password*.

> **Įspėjimas**
>
> Entering **username** and **password** in the *Authentication* tab will keep unprotected credentials in the connection configuration. Those **credentials will be visible** if, for instance, you shared the project file with someone. Therefore, it's advisable to save your credentials in an *Authentication configuration* instead (*Configurations* tab). See *Autentifikacijos sistema* for more details.

- HTTP *Referer*
- ☐ *Ignore GetCoverage URI reported in capabilities*: if checked, use given URI from the *URL* field above.
- ☐ *Ignore reported layer extents*: because the extent reported by raster layers may be smaller than the actual area which can be rendered (notably for WCS servers with symbology which takes more space than the data extent), check this option to avoid cropping raster layers to their reported extents, resulting in truncated symbols on the borders of these layers.

- ☐ *Ignore axis orientation*
- ☐ *Invert axis orientation*
- ☐ *Smooth pixmap transformation*

4. Press *OK* to create the connection.

Note that any proxy settings you may have set in your preferences are also recognized. Also note that it is possible to *Load* the connection parameters from a `.XML` file or *Save* them to a `.XML` file.

Now we are ready to load WCS layers from the above connection.

19.3 WFS and WFS-T Client

In QGIS, a WFS layer behaves pretty much like any other vector layer. You can identify and select features, and view the attribute table. QGIS supports WFS 1.0.0, 1.1.0, 2.0 and OGC API - Features (OAPIF), including editing (through WFS-T). QGIS also supports background download and progressive rendering, on-disk caching of downloaded features and version autodetection.

Layers of servers implementing OGC API - Features - Part 4: Create, Replace, Update and Delete can be turned into editing mode to allow creating, modifying and deleting features. Note that each created/modified/deleted feature requires a dedicated network request, so performance might suffer in case of simultaneous modification of hundreds or more features at a time.

In general, adding a WFS layer is very similar to the procedure used with WMS. There are no default servers defined, so you have to add your own. You can find WFS servers by using the *MetaSearch plugin* or your favourite web search engine. There are a number of lists with public URLs, some of them maintained and some not.

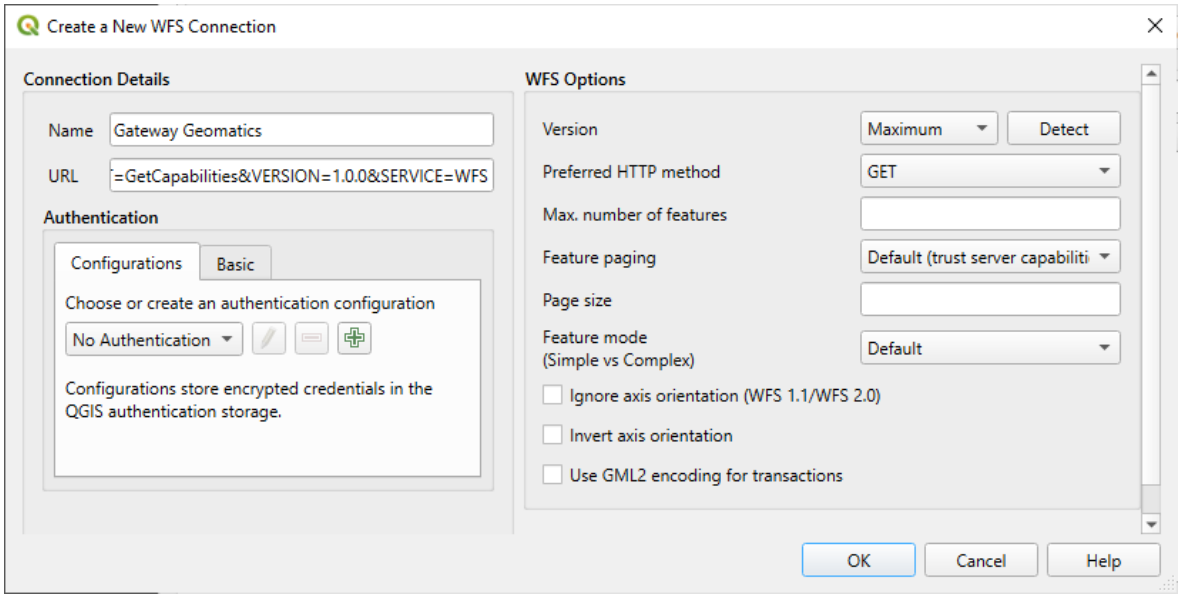
Loading a WFS Layer

As an example, we use the Gateway Geomatics WFS server and display a layer.

```
https://demo.gatewaygeomatics.com/cgi-bin/wfs_gateway?REQUEST=GetCapabilities&
↪VERSION=1.0.0&SERVICE=WFS
```

To be able to load a WFS Layer, first create a connection to the WFS server:

1. Open the *Data Source Manager* dialog by pressing the Open Data Source Manager button
2. Enable the *WFS / OGC API - Features* tab
3. Click on *New…* to open the *Create a New WFS Connection* dialog
4. In the *Connection Details* group, enter `Gateway Geomatics` as name
5. Enter the URL (see above)
6. Optionally, enter the *Authentication* settings



19.7 Fig. : Creating a connection to a WFS server

Pastaba

In case of an OGC API - Features (OAPIF), the URL to provide should be the landing page, i.e., the main page from which it is possible to navigate to all the available service endpoints.

7. In the *WFS Options* group, you can:
   - Indicate the WFS version of the server. If unknown, press the *Detect* button to automatically retrieve it.
   - Select the *Preferred HTTP method* to use for requests. The default is *GET*, but you can also select *POST*.
   - Define the *Maximum number of features* retrieved in a single GetFetFeature request. If empty, no limit is set.
   - *Feature mode (simple vs complex)*: helps determine how to handle servers serving data with simple or complex GML schemas. Available options are:
     - *Default*: if the server supports editing capabilities, this is the same as using *Simple Features*. If not, it fallbacks to using *Complex Features* mode.
     - *Simple Features*: returns data as simple features only
     - *Complex Features*: uses the *OGR GMLAS <https://gdal.org/en/stable/drivers/vector/gmlas.html>*, potentially returning complex features, depending on the data schema, where nested data structures are exposed by QGIS as JSON.
   - Depending on the WFS version, indicate whether to:
     - Enable *Feature paging* and specify the maximum number of features to retrieve with *Page size*. If no limit is defined, then the server default is applied.
     - Force to *Ignore axis orientation (WFS 1.1/WFS 2.0)*
     - *Invert axis orientation*.
     - *Use GML2 encoding for transactions*.

Įspėjimas

Entering **username** and **password** in the *Authentication* tab will keep unprotected credentials in the connection configuration. Those **credentials will be visible** if, for instance, you shared the project file with someone. Therefore, it's advisable to save your credentials in an *Authentication configuration* instead (*Configurations* tab). See *Autentifikacijos sistema* for more details.

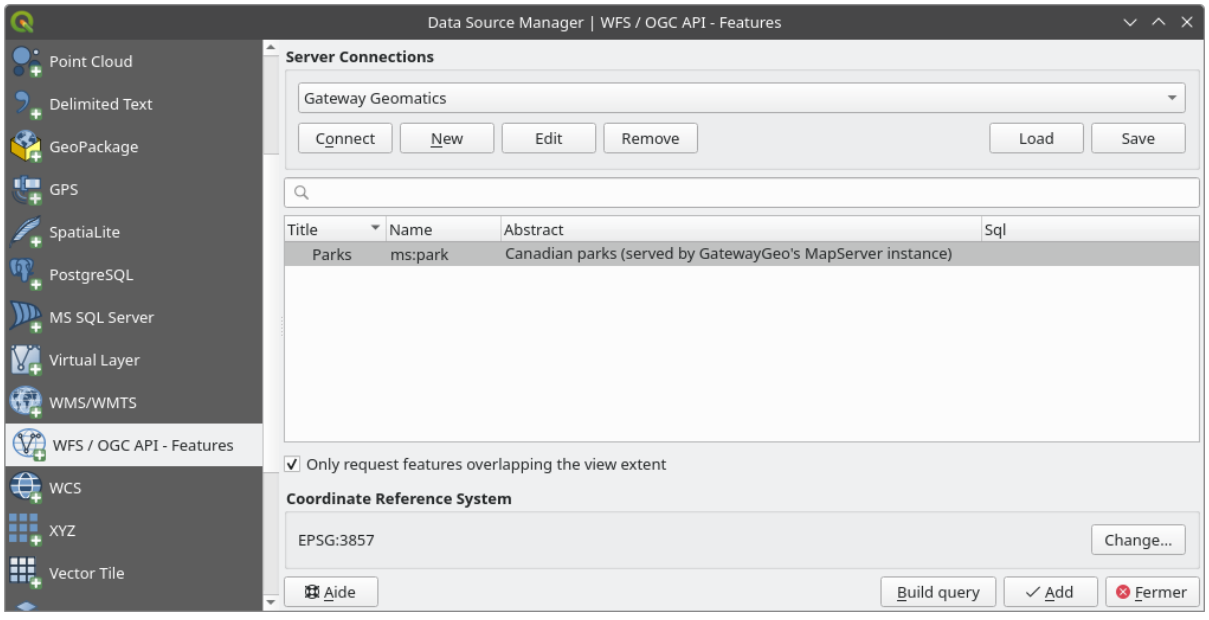
8. Press *OK* to create the connection.

Note that any proxy settings you may have set in your preferences are also recognized. Also note that it is possible to *Load* the connection parameters from a `.XML` file or *Save* them to a `.XML` file.

Now we are ready to load WFS layers from the above connection.

1. Choose ‚Gateway Geomatics' from the *Server Connections* drop-down list.
2. Click *Connect*
3. Select the *Parks* layer in the list
4. You can also choose whether to:
   - *Only request features overlapping the view extent*
   - *Change…* the layer's CRS to any other supported by the service
   - or build a query to specify particular features to retrieve from the service: double-click on the row of the layer to open the *SQL Query Composer* dialog. That dialog provides widgets to write an advanced SQL query relying on available tables and columns of the service, with sorting and filtering and a bunch of SQL functions, spatial predicates and operators.

     The query you build will appear after validation in the *SQL* column within the *WFS / OGC API - Features* table, and the filtered layer will display the icon next to it in the *Layers* panel. It is thus possibe to adjust the query at any moment.
5. Click *Add* to add the layer to the map, using its default name from the queried service.



19.8 Fig. : Adding a WFS layer

You'll notice the download progress is visualized in the lower left of the QGIS main window. Once the layer is loaded, you can identify and select a couple of features and view the attribute table.

19.4 SensorThings

QGIS supports connections to OGC SensorThings API, a standard providing an open and unified framework to interconnect IoT sensing devices, data, and applications over the Web. It is an open standard addressing the syntactic and semantic interoperability of the Internet of Things. It is based on the Observations and Measurements data model, a standardized model for observations, and for features involved in sampling when making observations.

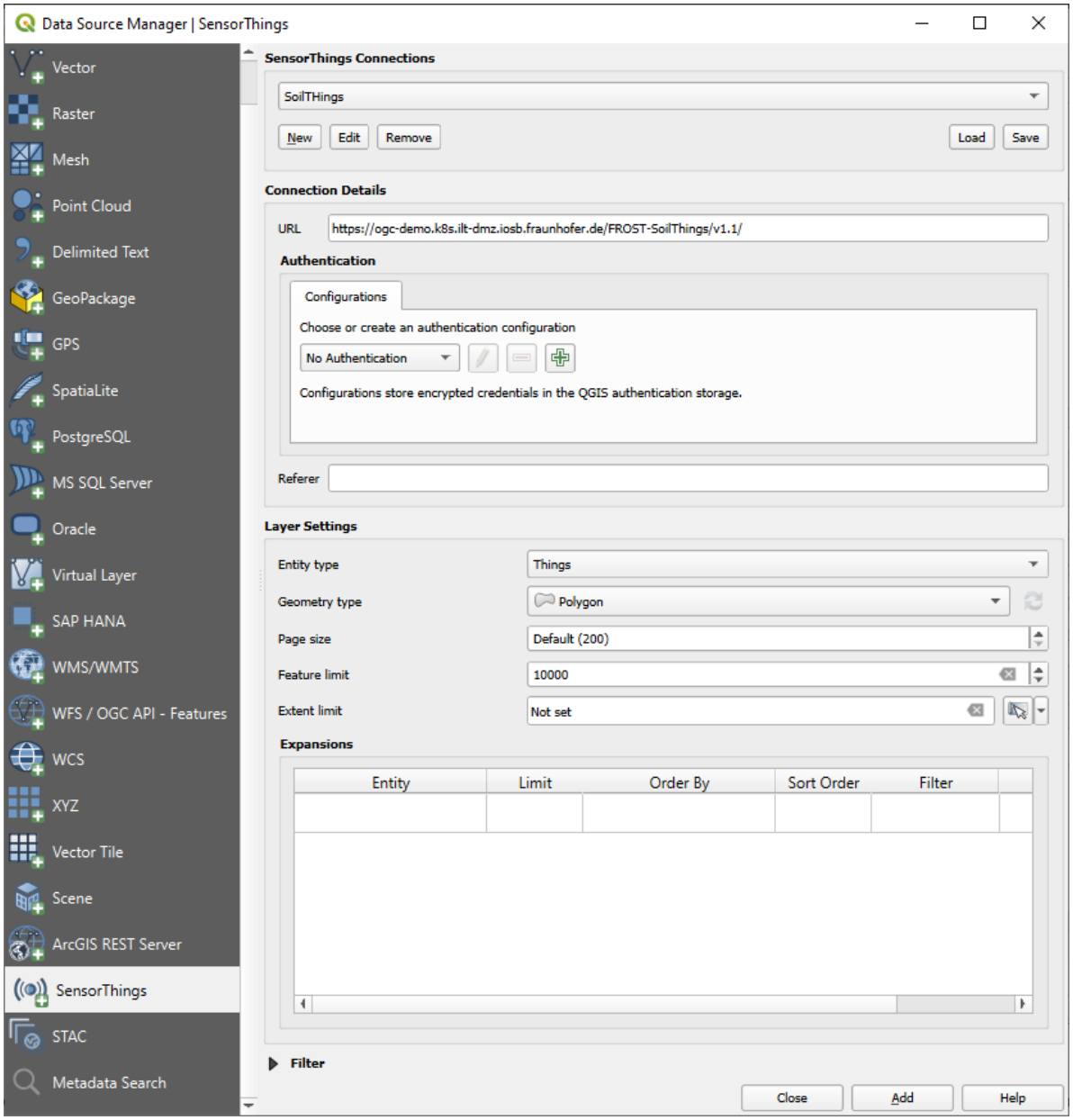
19.4.1 Setting connection

To add SensorThings data to QGIS use the *SensorThings* tab in the *Data Source Manager* dialog.

To establish a new connection, press *New* (or *New SensorThings Connection* from the Browser panel) and provide *Name* and *URL*. Advanced options, such as *authentication* and a *Referer*, can also be configured.

Press *OK* to establish the connection. Then you will be able to:

- *Edit* the SensorThings connection settings
- *Remove* the SensorThings connection

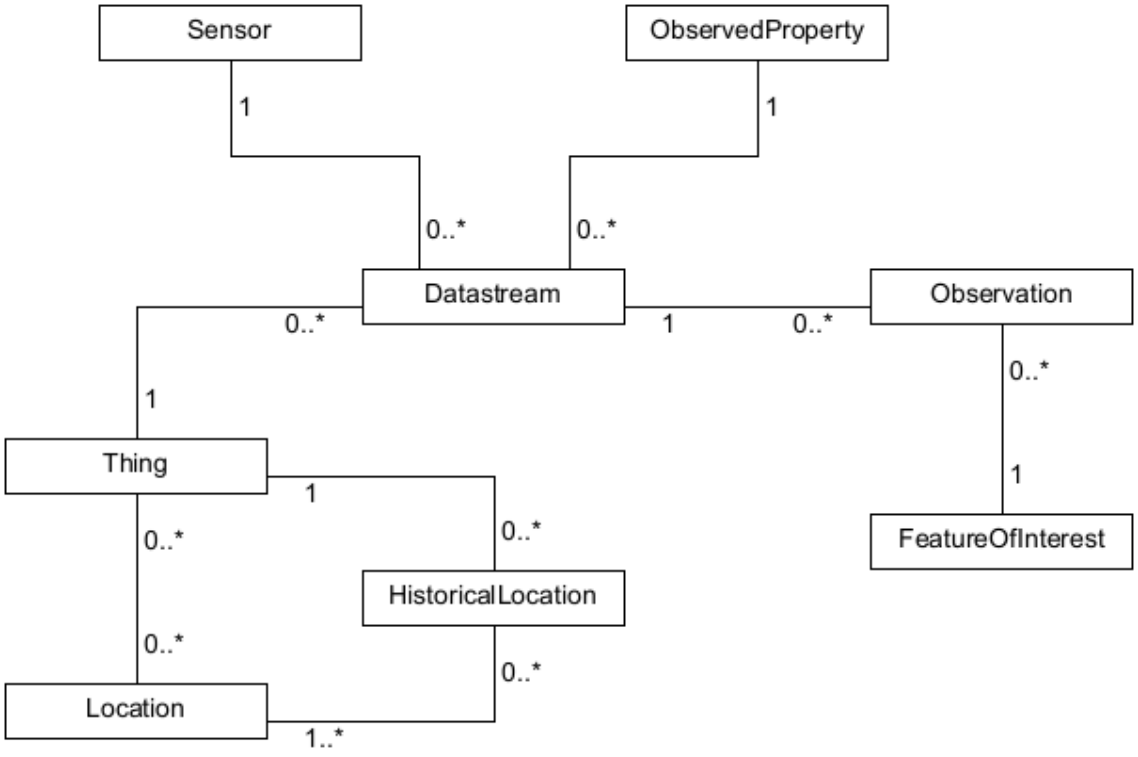


19.9 Fig. : SensorThings Connection dialog

Configurations can be saved to an `.XML` file (*Save*) through the *SensorThings* entry in *Data Source Manager* dialog or its contextual menu in the *Browser* panel (*Save Connections*). Likewise, configurations can be added from a file (*Load*).

19.4.2 Loading SensorThings data

Relations between layers (so-called entities) stored in a SensorThings dataset are expressed in the diagram below.



19.10 Fig. : Data model Observations and Measurements (Source: Sensor Web Tutorial by SIST network)

Any type of entity can be loaded in QGIS, but not all are spatial data. To load an entity, there are *Layer Settings* that can be configured:

- *Entity Type*: the entity to load from the data model as layer in QGIS
- *Geometry Type*: the geometry type of the selected entity to load. Press Check available types to limit the list to the actually supported geometry types.
- *Page Size*
- *Feature Limit* sets a maximum number of features to request from the service
- *Extent Limit* sets a maximum extent limit for the layer, so that only features within the extent are requested
- *Expansions*: The data model of SensorThings provides a mechanism of expansion of the results to related entities, similar to how tables are joined together in a relational database. Using this approach, you can expand the selected layer to include data from other items. This will flatten the relationship, creating as many parent features as children, and additional properties are added as columns in the attribute table.

Use *Filter* to build a query to filter the data, using SensorThings filter syntax.

Pastaba

The above settings and filtering options are also available for update in the layer properties dialog, *Source* tab, once loaded in QGIS.

Press *Add* to load the selected entity type as layer in QGIS.

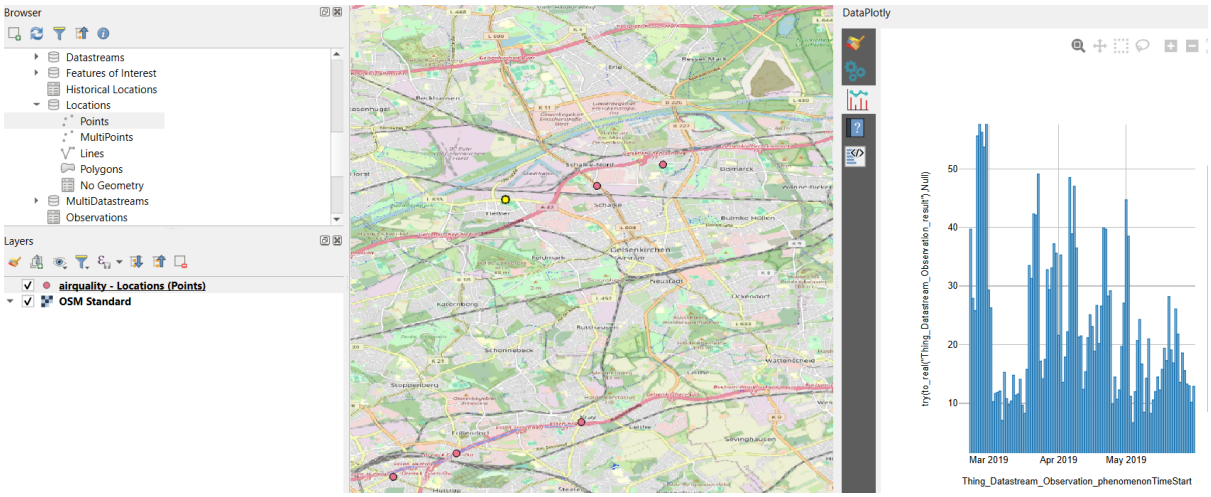
19.4.3 Working with a vector layer from SensorThings

A SensorThings layer is loaded in QGIS as a vector layer. As such, it displays the same tabs in the *layer properties* and allows same feature interactions using the selection or identify tools. There are however some specificities you should consider while working with SensorThings data.

Because of the data model, the result property of a SensorThings Observation is a string field. In case you want to use its numerical representation in for example a graduated style, use an expression to convert the value to real and try() in case this fails (e.g., `try( to_real("Observation_result"), Null)`).

In case you want to create a chart of the observations at one or more locations, you can install the QGIS plugin *Data Plotly*.

1. Now select the observations at a point location in the map view.
2. Open the plotly panel and activate the *Use only selected features* checkbox.
3. Select on the x-column a date-time property and on the y-column the *Observation_result*. This will plot the observations at that location over time.
4. Verify to filter by a single Observed Property.
5. Notice that the chart changes as soon as you select other locations on the map.



19.11 Fig. : Use Data plotly to plot the air quality observations at a location

19.5 STAC (SpatioTemporal Asset Catalogs)

STAC (SpatioTemporal Asset Catalog) is a specification for describing geospatial data in a consistent and accessible way. It defines a standard structure for organizing and indexing spatial-temporal assets such as satellite imagery, drone photos, or sensor data so that you can search, preview, and use these datasets across different platforms and tools.

STAC enables QGIS users to connect to and browse geospatial datasets, either from static catalogs or dynamic APIs, and download or stream assets such as imagery or other spatial data.

More information at https://stacspec.org/en.

Key components of STAC in QGIS include:

- **STAC Items**: The basic unit of a STAC catalog. Each item represents one geospatial asset or a group of related assets (e.g. image bands) at a specific time and location. It includes metadata such as geometry, datetime, links to assets (e.g., TIFFs, JSON, COG), and properties.
- **STAC Collections**: A grouping of items that share common characteristics and metadata (e.g. satellite mission). Collections may define spatial and temporal extents, licensing, and keywords.
- **STAC Catalogs**: A hierarchical container that organizes items and/or collections. Catalogs allow navigation of STAC datasets but do not necessarily include search capabilities.
- **STAC API**: An implementation of the STAC specification that allows querying and filtering of STAC items using spatial and temporal based filters, as well as searching within specific collections. STAC APIs follow the OGC API - Features pattern and support dynamic access to datasets.

There is an important distinction between static STAC catalogs and STAC API endpoints:

- **Static STAC Catalogs**: are collections of JSON files without search capabilities. They can be browsed via the Browser panel.
- **STAC API**: provide search capabilities and can be accessed in QGIS through both the Browser panel and the Data Source Manager.

19.5.1 Setting connection

STAC connections can be added in QGIS using either the *Browser panel* or the *Data Source Manager*:

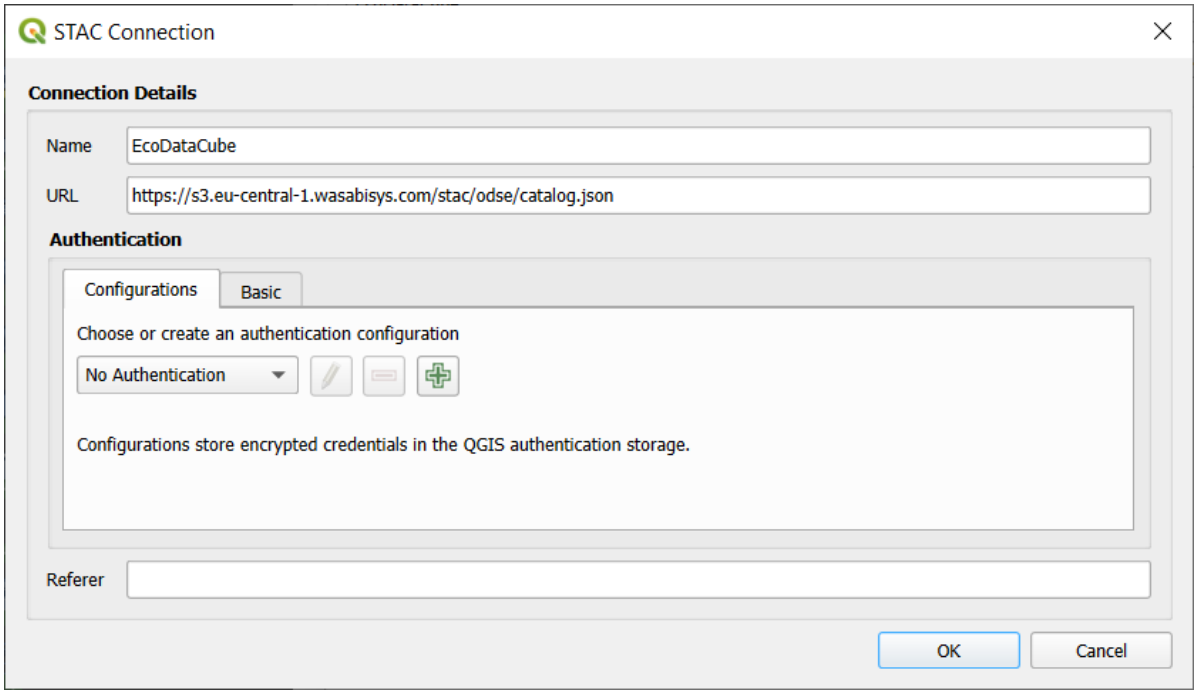
- **Browser Panel:** In the *Browser panel*, right-click on the *STAC* entry and select *New STAC Connection…*. In the dialog that appears, enter a *Name* for the connection, the *URL* of the STAC catalog and optionally fill in *Authentication* credentials and a *Referer*. Then click *OK*.

  Use this method for browsing static STAC catalogs that do not support search or filtering.

- **Data Source Manager:** For STAC APIs you can use *Data Source Manager* dialog.

  Open the *Data Source Manager*, choose the *STAC* tab and click the *New* button. Fill in the *Name* and *URL* fields, and (optional) the *Authentication* credentials and a *Referer*. Press *OK* and then *Connect* to establish the connection, after that you will be able to:

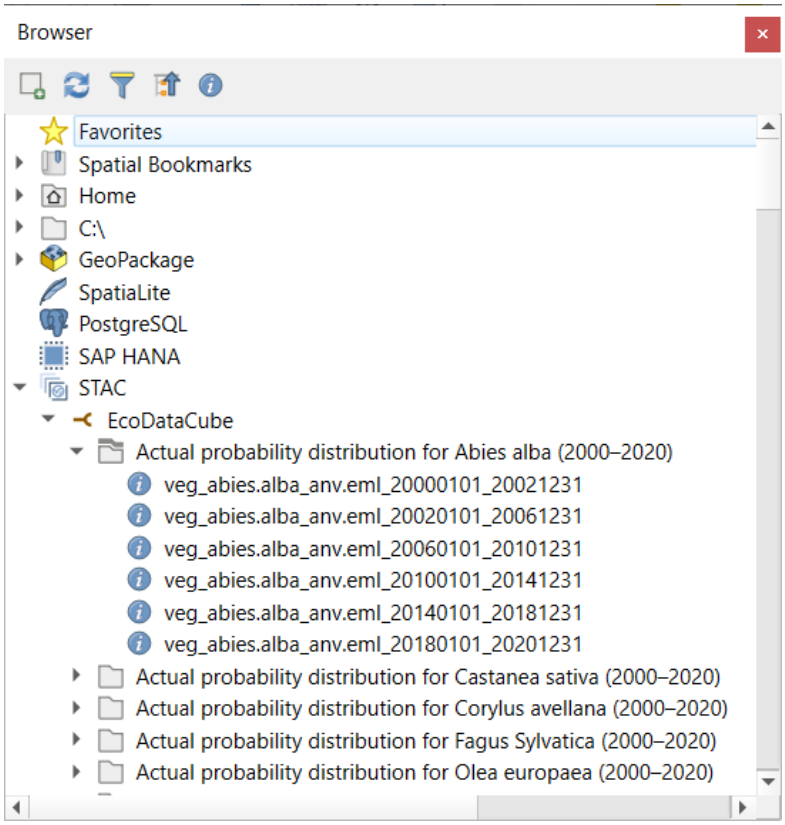
  - *Edit* the STAC connection settings
  - *Remove* the STAC connection



19.12 Fig. : Creating a connection to a STAC server

19.5.2 Browsing STAC Catalogs

Static STAC catalogs are displayed as hierarchical structures within the *Browser panel*. Once connected, the STAC catalog appears under *STAC* in the *Browser* panel. You can expand the catalog node to see its *Collections*. Expanding a collection reveals the individual *Items* it contains.



19.13 Fig. : STAC connection expanded in the Browser, showing Collections and Items

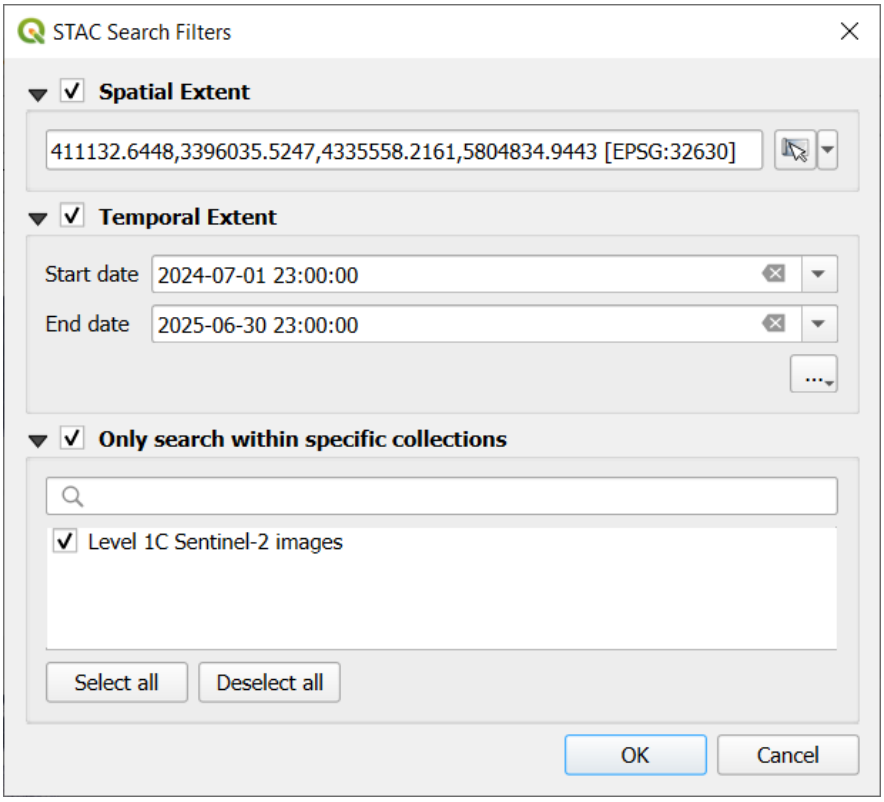
Right-click any STAC *Item* and choose *Details…* to view its metadata. The details panel shows the item's JSON content and a map of its coverage. If an item's asset is a cloud-optimized format (e.g. a `.COG`), you can add it directly to the map canvas. Otherwise, *Download Assets…* to save it locally before use.

19.5.3 Filtering and Searching STAC Items

For STAC API endpoints, QGIS supports spatial and temporal filtering, as well as limiting searches to specific collections.

To search for items:

1. Open *Data Source Manager* and go to the *STAC*
2. Select the STAC connection you created earlier and click *Filters…*



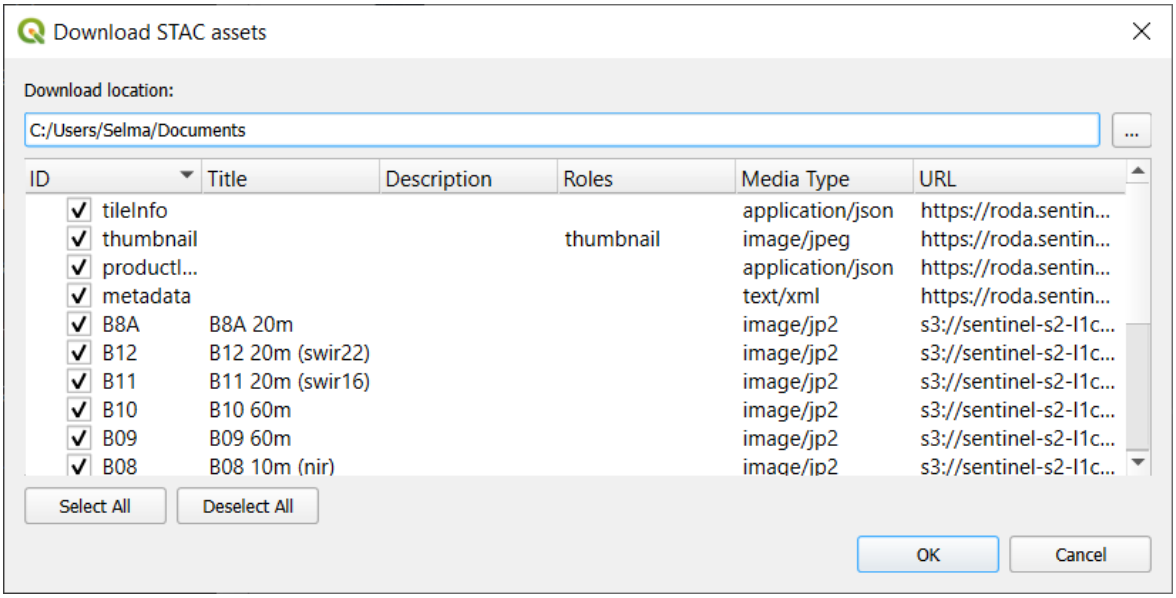
19.14 Fig. : Define spatial and temporal filters for a STAC catalog search

3. In the filter dialog, you can:
   - Enable *Spatial Extent* to restrict results to a bounding box or drawn area.
   - Enable *Temporal Extent* and define a date or range.
   - Limit to specific collections using *Only search within specific collections*
4. After applying filters, results will be listed in the main *Data Source Manager* panel.
5. Optionally check *Show Footprints* to display the footprints of the results on the map.

Right-click a result item to access actions:

- *Zoom to Item*
- *Pan to Item*
- *Download Assets*
- *Details…*

If the item's asset requires download, use the *Download Assets* option.



19.15 Fig. : Download STAC item assets

Downloaded assets include the main dataset and any auxiliary files such as thumbnails or style files. After download, use standard QGIS tools (e.g., *Add Raster Layer…*) to load and display the data.

19.6 ArcGIS REST Server

An ArcGIS REST Server can host many different types of web services (feature service, map service, image service, …). Read *ArcGIS REST serverių naudojimas* for instructions to connect to an ArcGIS REST server.

When loading layers form an ArcGIS Feature Service, symbols are automatically converted to QGIS-native symbology, allowing QGIS to visually reproduce the same appearance as ArcGIS web maps.

Layers from an ArcGIS Feature Service can be editable in QGIS if:

1. The ArcGIS service has editing enabled
2. Your credentials grant write access
3. And the remote layer supports editing operations (Create, Update, or Delete capabilities). If these options are disabled on the server, the layer will be read-only in QGIS.

Editing these layers follow the *same procedure* as for other vector layers. To update your map with the latest changes, use *View* ► *Refresh* or press `F5`. If multiple users edit the same layer concurrently, it is recommended to refresh your data before performing edits to avoid conflicts.

SKYRIUS 20

Working with GPS Data

20.1 Introducing GNSS/GPS Data

20.1.1 What is GPS?

GPS, the **Global Positioning System**, is a satellite-based system that allows anyone with a GPS receiver to find their exact position anywhere in the world. GPS is used as an aid in navigation, for example in airplanes, in boats and by hikers. The GPS receiver uses the signals from the satellites to calculate its latitude, longitude and (sometimes) elevation. Most receivers also have the capability to store:

- locations (known as **waypoints**)
- sequences of locations that make up a planned **route**
- and a **track** log of the receiver's movement over time.

Waypoints, routes and tracks are the three basic feature types in GPS data. QGIS displays waypoints in point layers, while routes and tracks are displayed in linestring layers.

> **Pastaba**
>
> QGIS supports also GNSS receivers. But we keep using the term GPS in this documentation.

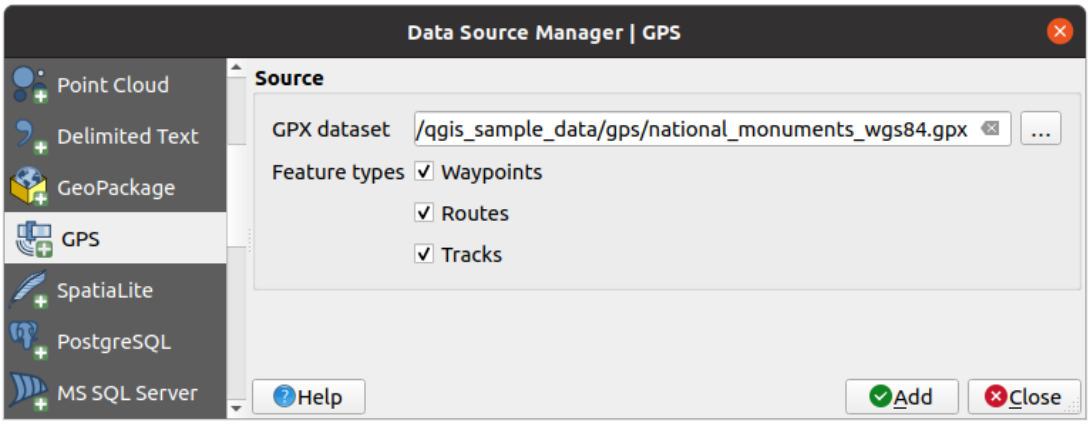
20.1.2 Transferring or loading GPS data

Loading a GPX file

Yra daug formatų, leidžiančių laikyti GPS duomenis. QGIS naudoja formatą, vadinamą GPX (GPS eXchange format), kuris yra standartinis apsikeitimo formatas, kuriame gali būti bet koks skaičius taškų, maršrutų ir pėdsakų viename faile.

To load a `GPX` file:

1. Open the *GPS* tab in the *Data Source Manager* dialog, i.e.:
   - Click the Open Data Source Manager button on the toolbar (or press `Ctrl+L`) and enable the target tab
   - Or select *Layer* ► *Add Layer* ► *Add GPX Layer…*
2. Use the … Browse button next to the *GPX dataset* option to select the GPX file
3. Use the check boxes to select the *Feature types* you want to load from the file. Each feature type (*Waypoints*, *Tracks* or *Routes*) will be loaded in a separate layer.

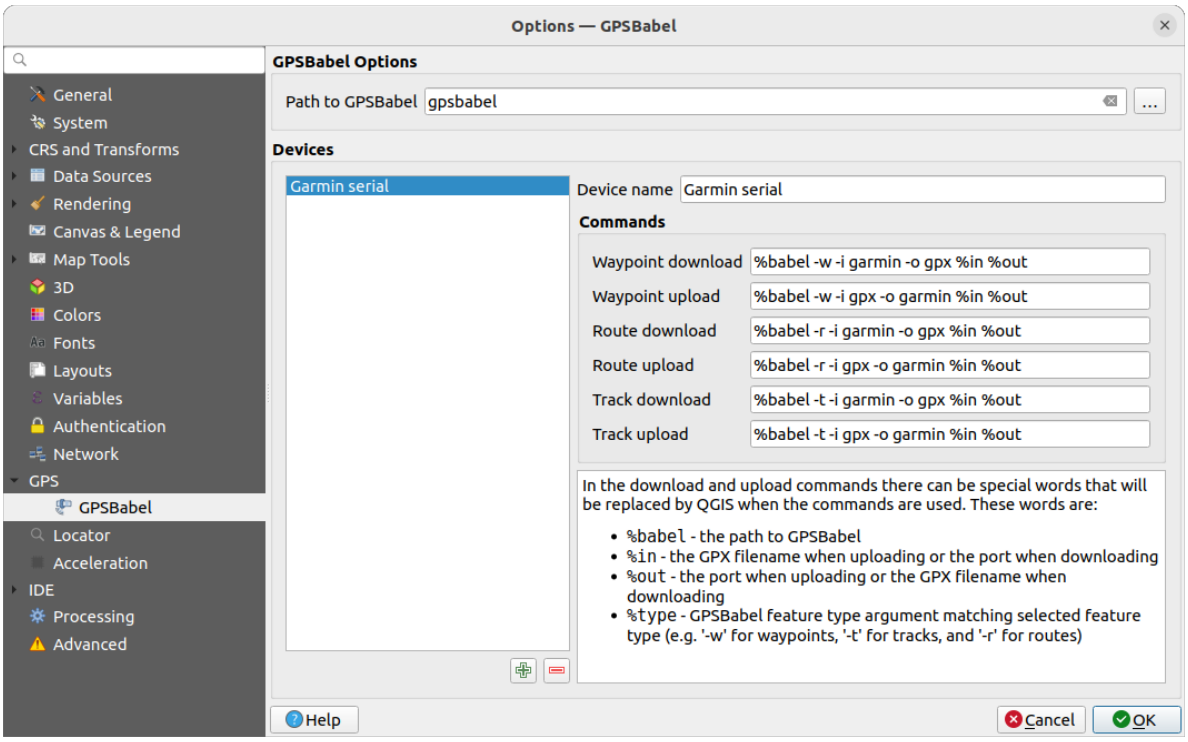


20.1 Fig. : GPS duomenų įkėlimo dialogas

Loading to or from a device

There are lots of different types of GPS devices and formats. Since QGIS uses GPX files, you need a way to convert other GPS file formats to GPX. QGIS can do that using the free program GPSBabel. GPSBabel can help you convert waypoints, tracks, and routes between popular GPS receivers such as Garmin or Magellan and mapping programs like Google Earth or Basecamp. Literally hundreds of GPS receivers and programs are supported. It can also transfer GPS data between your computer and a GPS device.

Under *Settings* ► *Options* ► *GPS* ► *GPSBabel*, QGIS allows you to define your own device type and set parameters of conversion that could later be used by the *Processing GPS algorithms*.



20.2 Fig. : GPS Babel nustatymai

1. Iš pradžių turite nurodyti *kelią iki GPSBabel* programos.
2. Tada galite pridėti jūsų įrenginį. Galite atnaujinti įrenginių sąrašą, naudodami mygtukus symbologyAddl Pridėti naują įrenginį ar Išimti įrenginį.
3. Kiekvienam įrenginiui:
   - nurodote *Įrenginio pavadinimą*
   - sukonfigūruojate skirtingas *komandas*, kurias QGIS naudos bendravimui su įrenginiu, tokias kaip:
     - *Atsisiųsti taškus* iš įrengnio
     - *Įkelti taškus* į įrenginį
     - *Atsisiųsti maršrutą* iš įrenginio
     - *Įkelti maršrutą* į įrenginį
     - *Atsisiųsti pėdsaką* iš įrenginio
     - *Įkelti pėdsaką* į įrenginį

     Nors komandos paprastai yra GPSBabel komandos, jūs taipogi galite naudoti bet kokią komandinės eilutės programą, kuri sukuria GPX failą. QGIS pakeis raktažodžius `%type`, `%in` ir `%out` komandos vykdymo metu.

     As an example, if you create a device type with the download command `gpsbabel %type -i garmin -o gpx %in %out` and then use it to download waypoints from port `/dev/ttyS0` to the file `output.gpx`, QGIS will replace the keywords and run the command `gpsbabel -w -i garmin -o gpx /dev/ttyS0 output.gpx`.

     Visas komandinės eilutės parinktis, kurių gali prireikti jūsų atvejui, rasite GPSBabel vadove.

Sukūrus naują įrenginio tipą jis atsiras įrenginių sąrašuose GPS atsiuntimo ir nusiuntimo algoritmuose.

Pastaba

GPS units allow you to store data in different coordinate systems. When downloading a GPX file (from your GPS unit or a web site) and then loading it in QGIS, be sure that the data stored in the GPX file uses WGS 84 (latitude/longitude). QGIS expects this, and it is the official GPX specification. See GPX 1.1 Schema Documentation.

20.2 Live GPS tracking

QGIS can help you do field mapping with a GPS receiver. Such a live tracking operation is done using the *GPS Toolbar*. Some *device configuration* may be needed before connecting QGIS and the GPS receiver.

20.2.1 GPS toolbar

The *GPS Toolbar* provides the main tools to control a live tracking session. It can be activated from *View* ► *Toolbars* ► *GPS*. It follows the state of the project, GPS, and current GPS track, and enables actions only when they make sense. Once a device is detected by QGIS, you will be able to interact with it:

- Connect to GPS
- Disconnect from GPS
- Recenter map on GPS location: the map will immediately recenter on the current GPS position regardless of the *automatic recentering parameter*.
- Set destination layer for GPS digitized features: By default, QGIS will *Follow active layer* for feature digitizing, meaning that GPS digitizing tools will adapt to the layer selected in *Layers* panel and created features will be stored in it accordingly. While this may be useful in some contexts, it also implies careful interaction with other layers to avoid storing inadvertently features in undesired layers. This option allows you to explicitly indicate a layer for the data storage during the live tracking session, and you could switch depending on your needs. Exposed tools will adapt as well to the selected layer type.

  The GPS destination layer is made automatically editable when a feature is being created, and the user is notified accordingly.
- actions for feature creation:
  - Add vertex to GPS track using GPS current location
  - Create a new point feature at current GPS location
  - Create a new line feature using the current GPS track
  - Create a new polygon feature using the current GPS track
  - Reset track
- Show GPS information: opens the *GPS Information* panel
- a display box for quick access to some GPS information components you may need:
  - Show location
  - Show altitude (Geoid)
  - Show altitude (WGS 84 ellipsoid)

  - ☐ Show ground speed
  - ☐ Show bearing
  - ☐ Show total track length
  - ☐ Show distance from start of track

- The 🔧 Settings button opens a drop-down menu with common settings which are expected to be modified mid-session:
  - ☐ *Show location marker*
  - ☐ *Show bearing line*
  - ☐ *Rotate map to match GPS direction*
  - Options to control map recentering:
    - ○ *Always recenter map* when the GPS position is offset from the map canvas center by a certain distance (as a ratio of the map canvas extent)
    - ◉ *Recenter map when leaving extent*
    - ○ *Never recenter*
  - ☐ *Automatically add track vertices* whenever new locations are received from the GPS device.
  - ☐ *Automatically save added features*: features created from GPS locations are immediately committed to their target layers (skipping the usual layer edit buffer)
  - *Time stamp destination* to adjust the field for time fix storage
  - ☐ *Log to GeoPackage/SpatiaLite…*: When activated, the user will be prompted to select an existing GeoPackage/SpatiaLite file or enter a new file name. A `gps_points` and `gps_tracks` table will be created in the file with a predefined structure.

    All incoming GPS messages will be logged to the `gps_points` layer, along with speed, bearing, altitude and accuracy information from the GPS.

    When the GPS is disconnected (or QGIS closed), the entire recorded GPS track will be added to the `gps_tracks` table (along with some calculated information like track length, start and end times).
  - ☐ *Log NMEA sentences…*: enables logging all raw NMEA strings from the device to a text file
  - 🔧 *GPS settings…* to access the GPS *global options* dialog

Patarimas

Live status bar information on position

When a GPS device is connected and the user moves the cursor over the map canvas, a live status bar message displays the distance and bearing from the cursor to the GPS position. Project distance and bearing settings are respected in this display. On a touch screen device, use a tap-and-hold event to trigger the live status bar message.

20.2.2 GPS Information Panel

For full monitoring of a live GPS tracking in QGIS, you may need to enable the *GPS Information Panel* (*View* ► *Panels* ► or press `Ctrl+0`).

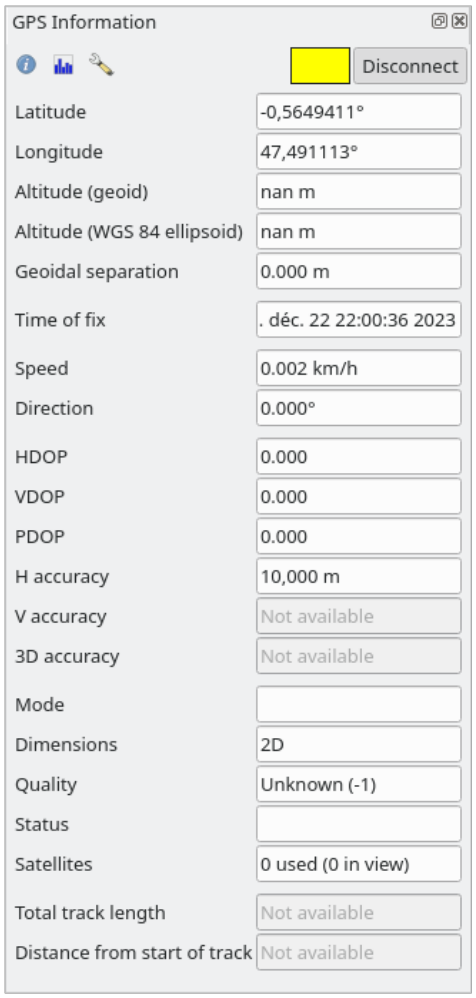
In the top right corner of the *GPS Information Panel*, you press *Connect* to initiate connection between QGIS and a plugged-in GPS receiver, or *Disconnect* them.

In the top left of the panel, following buttons are accessible:

- Position: live details on the GPS position and sensors
- Signal: signal strength of satellite connections
- Settings: drop-down menu to *live tracking options* you may need to modify during a session

Position and additional attributes

In the Position tab, if the GPS is receiving signals from satellites, you will see your position in latitude, longitude and altitude together with additional attributes.



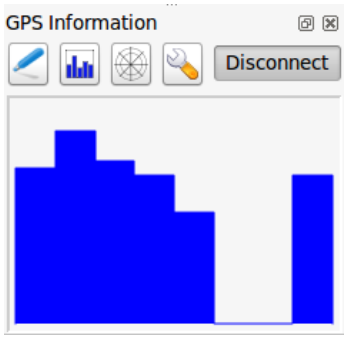
20.3 Fig. : GPS tracking position and additional attributes

- *Latitude*
- *Longitude*

- *Altitude (Geoid)*: Altitude/elevation above or below the mean sea level
- *Altitude (WGS 84 ellipsoid)*: Altitude/elevation above or below the WGS-84 Earth ellipsoid
- *Geoidal separation*: Difference between the WGS-84 Earth ellipsoid and mean sea level (geoid), − means mean sea level is below ellipsoid
- *Time of fix*
- *Speed*: Ground speed
- *Direction*: Bearing measured in degrees clockwise from true north to the direction of travel
- *HDOP*: Horizontal dilution of precision
- *VDOP*: Vertical dilution of precision
- *PDOP*: Dilution of precision
- *H accuracy*: Horizontal accuracy in meters
- *V accuracy*: Vertical accuracy in meters
- *3D accuracy*: 3D Root Mean Square (RMS) in meters
- *Mode*: GPS receiver configuration 2D/3D mode, can be `automatic` or `manual`
- *Dimensions*: Position fix dimension, can be `2D`, `3D` or `No fix`
- *Quality*: Positioning quality indicator
- *Status*: Position fix status, can be `Valid` or `Invalid`
- *Satellites*: Count of satellites used in obtaining the fix
- *Total track length*: Total distance of current GPS track
- *Distance from start of track*: Direct distance from first vertex in current GPS track to latest vertex

Signal

With Signal tab, you can see the signal strength of the satellites you are receiving signals from.



20.4 Fig. : GPS tracking signal strength

20.2.3 Connect to a Bluetooth GPS for live tracking

With QGIS you can connect a Bluetooth GPS for field data collection. To perform this task you need a GPS Bluetooth device and a Bluetooth receiver on your computer.

At first you must let your GPS device be recognized and paired to the computer. Turn on the GPS, go to the Bluetooth icon on your notification area and search for a New Device.

On the right side of the Device selection mask make sure that all devices are selected so your GPS unit will probably appear among those available. In the next step a serial connection service should be available, select it and click on *Configure* button.

Remember the number of the COM port assigned to the GPS connection as resulting by the Bluetooth properties.

After the GPS has been recognized, make the pairing for the connection. Usually the authorization code is `0000`.

Now open *GPS information* panel and switch to GPS options screen. Select the COM port assigned to the GPS connection and click the *Connect*. After a while a cursor indicating your position should appear.

If QGIS can't receive GPS data, then you should restart your GPS device, wait 5-10 seconds then try to connect again. Usually this solution work. If you receive again a connection error make sure you don't have another Bluetooth receiver near you, paired with the same GPS unit.

20.2.4 Examples of GPS devices connection in QGIS

Using GPSMAP 60cs

MS Windows

Easiest way to make it work is to use a middleware (freeware, not open) called GPSGate.

Launch the program, make it scan for GPS devices (works for both USB and BT ones) and then in QGIS:

1. In *Settings* ► *Options* ► *GPS*, detect the plugged-in device. You can use the *Autodetect* mode.
2. Press *Connect* in the *GPS Information* panel

Ubuntu/Mint GNU/Linux

As for Windows the easiest way is to use a server in the middle, in this case GPSD, so

1. Install the program

```
sudo apt install gpsd
```

2. Then load the `garmin_gps` kernel module

```
sudo modprobe garmin_gps
```

3. And then connect the unit and check with `dmesg` the actual device being used by the unit, for example `/dev/ttyUSB0`.
4. Now you can launch gpsd

```
gpsd /dev/ttyUSB0
```

5. And finally connect with the QGIS live tracking tool.

Using BTGP-38KM datalogger (only Bluetooth)

Using GPSD (under Linux) or GPSGate (under Windows) is effortless.

Using BlueMax GPS-4044 datalogger (both BT and USB)

MS Windows

The live tracking works for both USB and BT modes, by using GPSGate or even without it, just use the *Autodetect* mode, or point the tool to the right port.

Ubuntu/Mint GNU/Linux

For USB

The live tracking works both with GPSD

```
gpsd /dev/ttyACM3
```

or without it, by connecting the QGIS live tracking tool directly to the device (for example `/dev/ttyACM3`).

For Bluetooth

The live tracking works both with GPSD

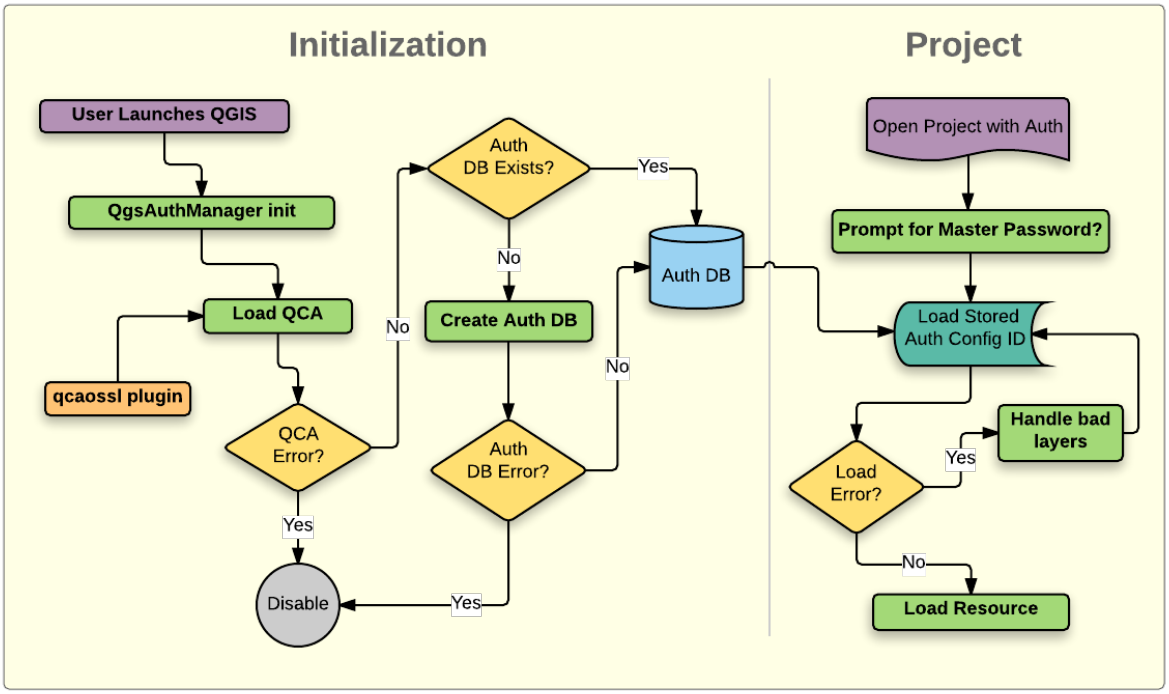
```
gpsd /dev/rfcomm0
```

or without it, by connecting the QGIS live tracking tool directly to the device (for example `/dev/rfcomm0`).

SKYRIUS 21

Autentifikacijos sistema

21.1 Authentication System Overview



21.1 Fig. : Anatomy of authentication system

21.1.1 Authentication database

The authentication system stores authentication configurations in an SQLite database file located, by default, at `<profile directory>/qgis-auth.db`.

This authentication database can be moved between QGIS installations without affecting other current QGIS user preferences, as it is completely separate from normal QGIS settings. A configuration ID (a random 7-character alphanumeric string) is generated when initially storing a configuration to the database. This represents the configuration, thereby allowing the ID to be stored in plain text application components, (such as project, plugin, or settings files) without disclosure of its associated credentials.

Pastaba

The parent directory of the *qgis-auth.db* can be set using the following environment variable, `QGIS_AUTH_DB_DIR_PATH`, or set on the command line during launch with the `--authdbdirectory` option.

21.1.2 Custom authentication databases

QGIS can be configured to use a custom authentication database instead of the above mentioned default SQLite one: any database suppported by the Qt SQL module can be used (e.g. PostgreSQL, MySQL, etc), provided that the corresponding Qt SQL driver is available in the system.

This can be useful in scenarios where a user wants to share the same authentication database between multiple QGIS installations, or when a user wants to use a different authentication database than the default SQLite one or when a centralized authentication database is used by QGIS server.

The only way to configure a custom authentication database is by setting the `QGIS_AUTH_DB_URI` environment variable to the URI of the connection, the URI is in the form of `driver://username:password@hostname:port/database?options`.

Where:

- `driver` is the name of the Qt SQL driver to use, e.g. `QPSQL` for PostgreSQL, `QMYSQL` for MySQL, etc.
- `username` is the username to use to connect to the database
- `password` is the password to use to connect to the database
- `hostname` is the hostname of the database server
- `port` is the port of the database server
- `database` is the name of the database to use
- `options` are the options to pass to the driver, e.g. `sslmode=require` for PostgreSQL

Pastaba

The `schema` can be specified in the URI options, e.g. `QPSQL://username:password@hostname:port/database?schema=schema_name`

The database must exist before starting QGIS, and the user must have the necessary permissions to connect to the database and to create the required tables if they do not exist.

Įspėjimas

The password in the URI is stored in plain text in the environment variable, so it is recommended to use a passwordless user or a user with limited permissions to connect to the database.

Įspėjimas

Any database not based on SQLite is considered to be read-only (this can be changed by Python plugins if necessary).

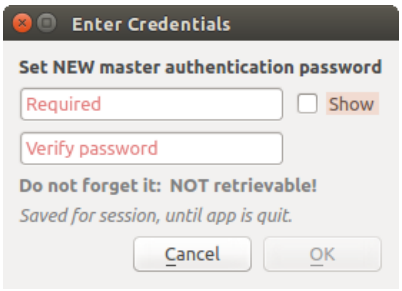
This is an advanced feature, designed to allow one or more custom authentication databases or even custom Python implementations of credentials storages to be used by QGIS.

The system is also designed to allow for multiple authentication databases to be used but there is currently no user facing interface to manage multiple credential storages so its usage requires manual configuration and management, typically from a Python plugin.

21.1.3 Master password

To store or access sensitive information within the database, a user must define a *master password*. A new master password is requested and verified when initially storing any encrypted data to the database. When sensitive information is accessed, the user is prompted for the master password. The password is then cached for the remainder of the session (until application is quit), unless the user manually chooses an action to clear its cached value. Some instances of using the authentication system do not require input of the master password, such as when selecting an existing authentication configuration, or applying a configuration to a server configuration (such as when adding a WMS layer).

You can choose to save the password in the `Wallet/Keyring` of your computer.



21.2 Fig. : Input new master password

Pastaba

A path to a file containing the master password can be set using the following environment variable, `QGIS_AUTH_PASSWORD_FILE`.

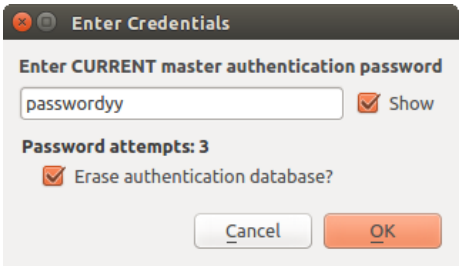
Managing the master password

Once set, the master password can be reset; the current master password will be needed prior to resetting. During this process, there is an option to generate a complete backup of the current database.

21.3 Fig. : Resetting master password

If the user forgets the master password, there is no way to retrieve or override it. There is also no means of retrieving encrypted information without knowing the master password.

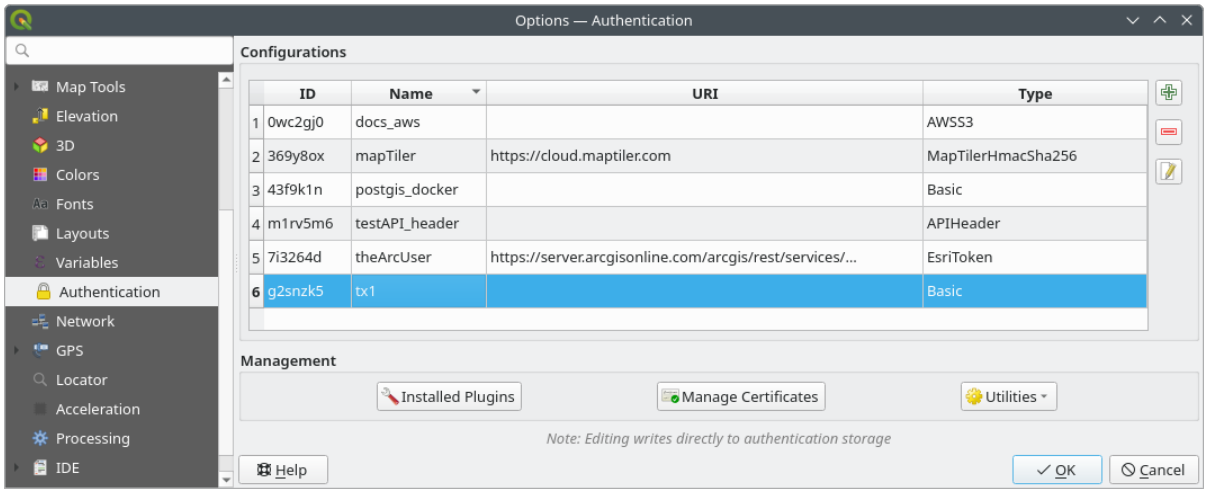
If a user inputs their existing password incorrectly three times, the dialog will offer to erase the database.



21.4 Fig. : Password prompt after three invalid attempts

21.1.4 Authentication Configurations

You can manage authentication configurations from *Configurations* in the *Authentication* tab of the QGIS Options dialog (*Settings* ► *Options*).

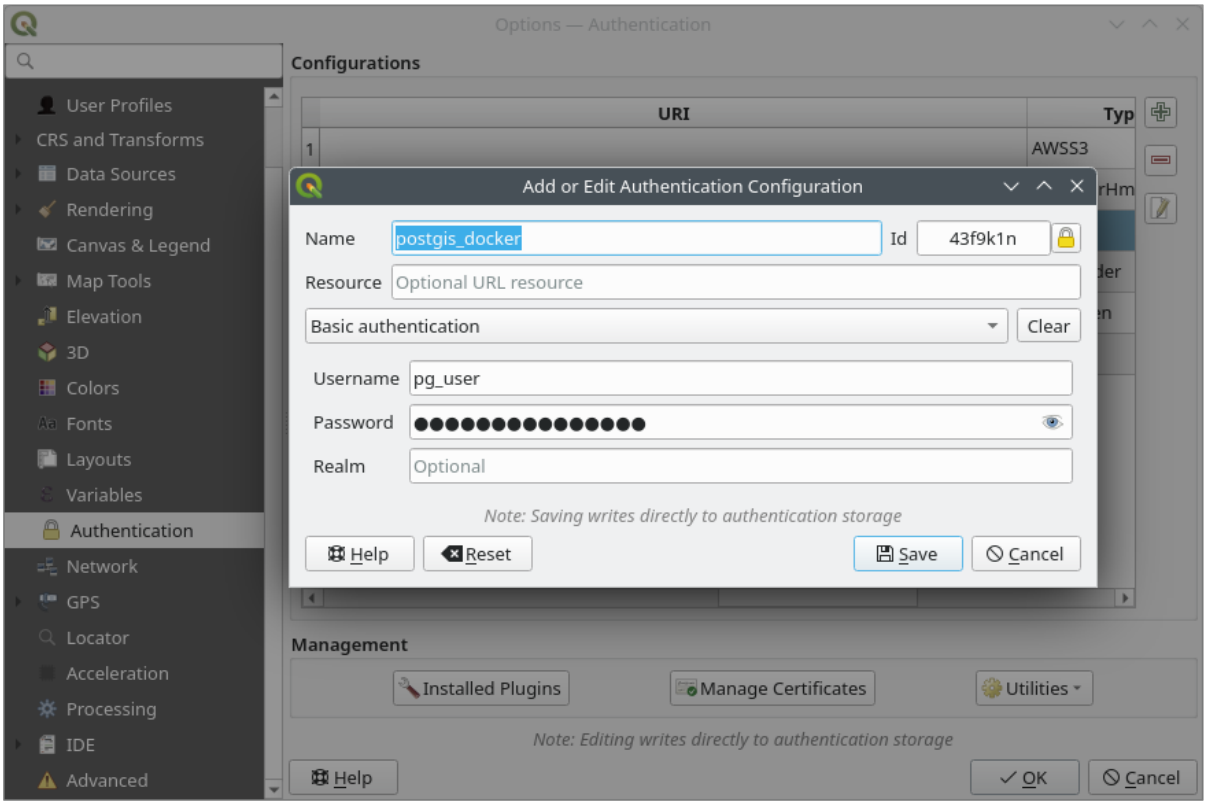


21.5 Fig. : Configurations editor

To create a new authentication configuration:

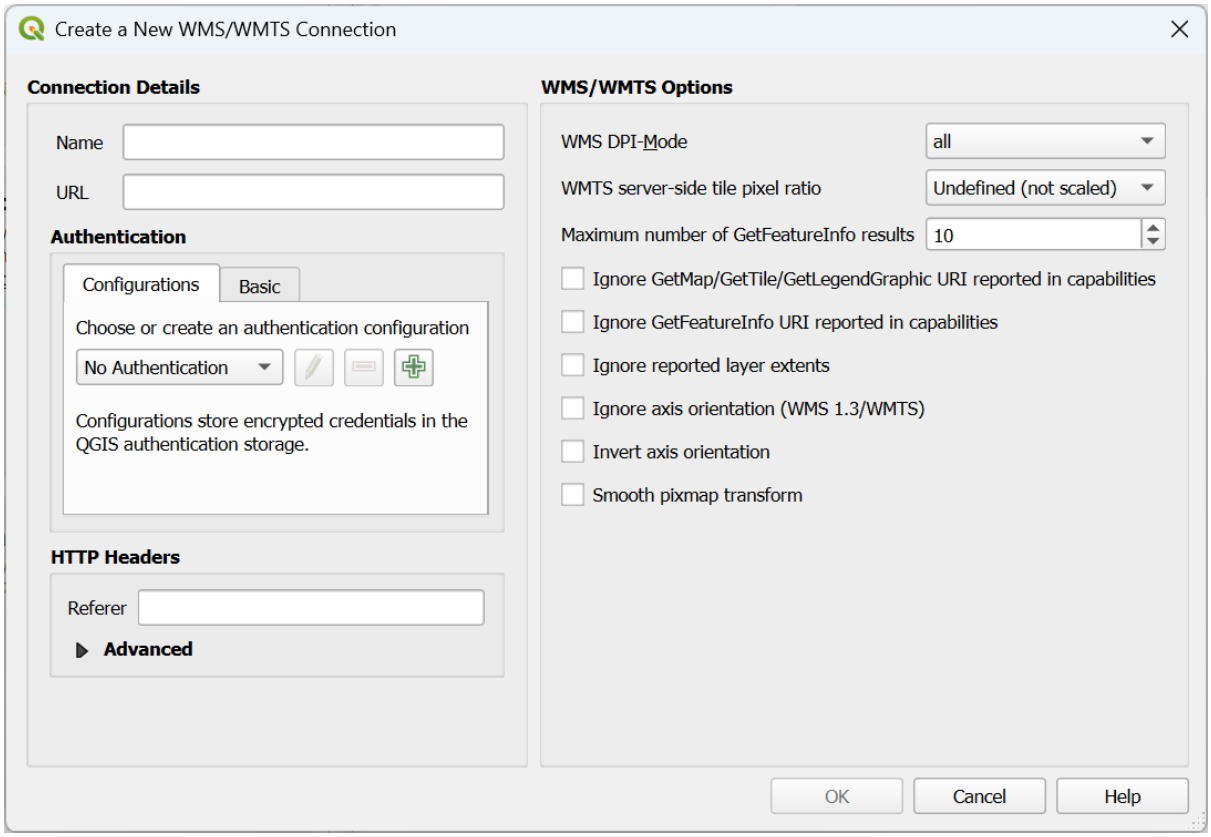
1. Press the Add a new authentication configuration
2. Provide a *Name* and optional *Resource* URL
3. A seven-characters *ID* will be assigned to the configuration for identification purpose. It can be customized after you pressed the Unlock to edit ID button. Be aware that updating that ID may break things.
4. Select the *target method* and fill the corresponding connection details
5. Once done, click *Save* to save the configuration. The saved configuration appears in the list of authentication configurations and can be used with supported connections.

The saved configuration can later be edited using Edit selected configuration or removed using Delete selected configuration.



21.6 Fig. : Adding config from within Configuration editor

The same type of operations for authentication configuration management (Add, Edit and Remove) can be done when configuring a given service connection, such as configuring an OWS service connection. For that, there are action buttons within the *configuration selector* for fully managing configurations found within the authentication database. In this case, there is no need to go to the *configurations* in *Authentication* tab of QGIS options unless you need to do more comprehensive configuration management.

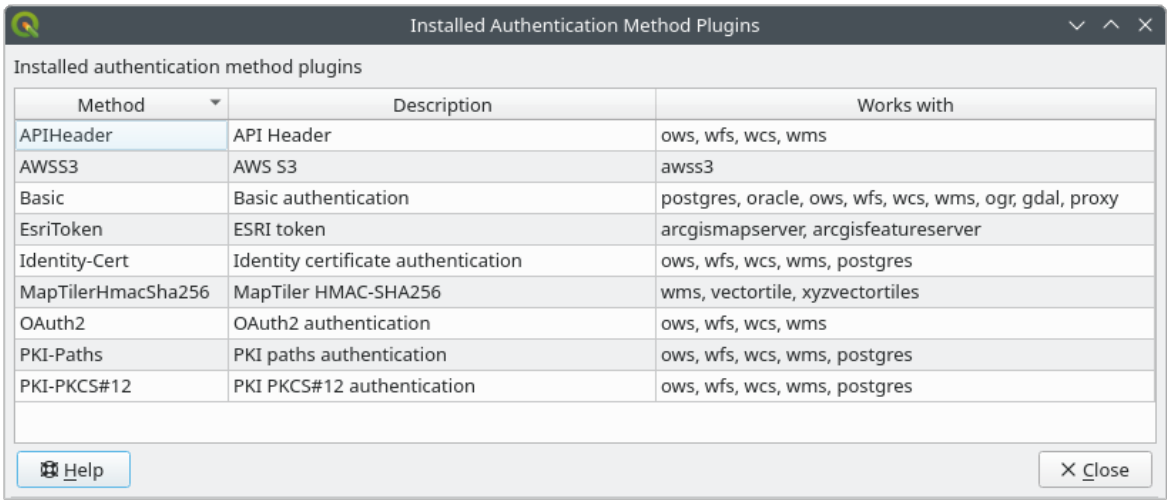


21.7 Fig. : WMS connection dialog showing authentication configuration widget

When creating or editing an authentication configuration, the info required is a name, an authentication method and any other info that the authentication method requires (see more about the available authentication types in *Authentication Methods*).

21.1.5 Authentication Methods

Available authentications are provided by C++ plugins much in the same way data provider plugins are supported by QGIS. The method of authentication that can be selected is relative to the access needed for the resource/provider, e.g. HTTP(S) or database, and whether there is support in both QGIS code and a plugin. As such, some authentication method plugins may not be applicable everywhere an authentication configuration selector is shown. A list of available authentication method plugins and their compatible resource/providers can be accessed going to *Settings* ► *Options* and, in the *Authentication* tab, click the *Installed Plugins* button.



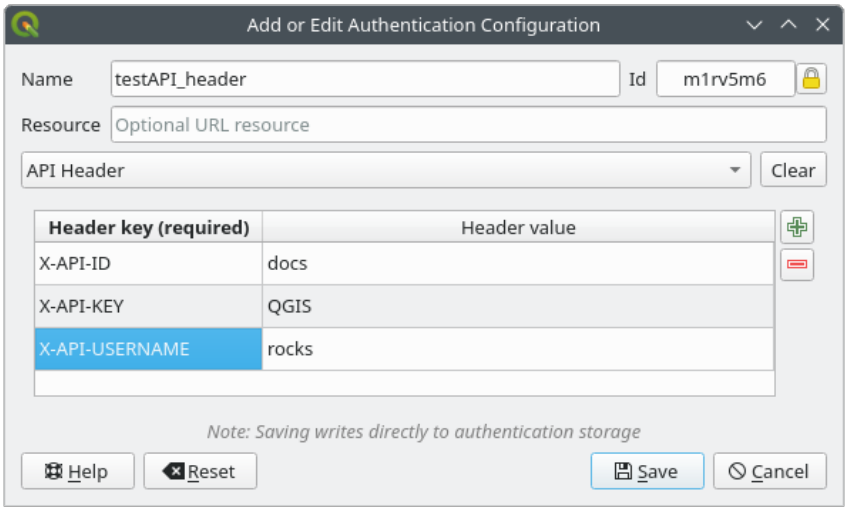
21.8 Fig. : Available method plugins list

Plugins can be created for new authentication methods that do not require QGIS to be recompiled. Since the support for plugins is currently C++-only, QGIS will need to be restarted for the new dropped-in plugin to become available to the user. Ensure your plugin is compiled against the same target version of QGIS if you intend to add it to an existing target install.

API Header Authentication

The API Header authentication method allows you to connect to services that require custom HTTP headers, such as API keys. For example, if a WMS service requires an API key sent as a header (e.g. `X-API-KEY`), you can use this method to provide it.

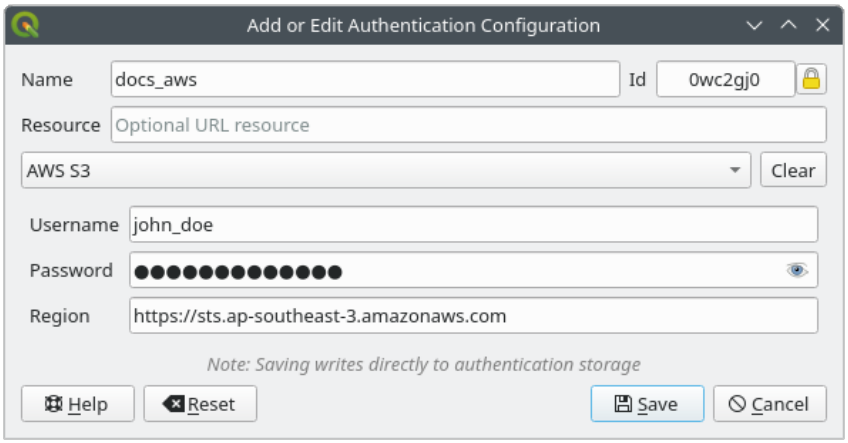
You can add one or more headers with the button; provide a *Header Name* and *Header Value* for each entry. Use to delete the selected header or the *Clear* button to remove all entries.



21.9 Fig. : API Header authentication configs

AWS S3 Authentication

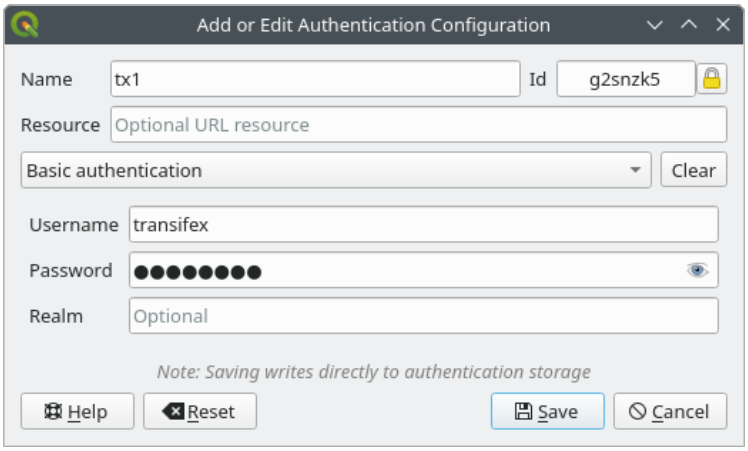
The AWS S3 authentication method allows you to connect to Amazon S3 resources by providing a *Name*, *Resource*, *Username*, *Password*, and *Region*.



21.10 Fig. : AWS S3 authentication configs

Basic Authentication

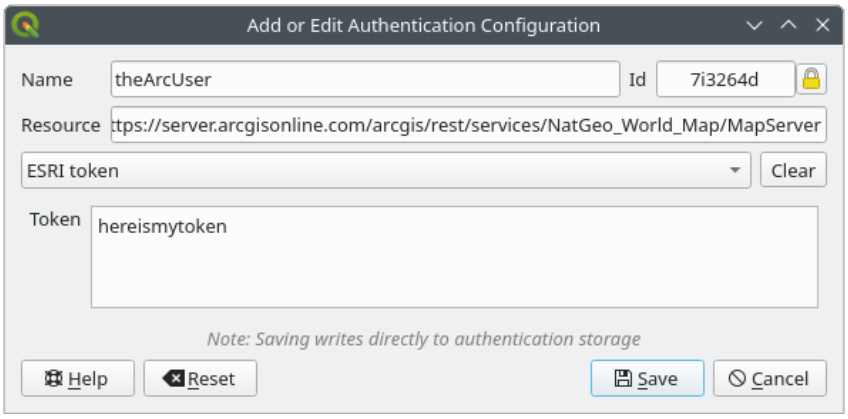
The Basic authentication method is used for services that require standard HTTP authentication with a *Username*, *Password* and *Realm*.



21.11 Fig. : Basic HTTP authentication configs

ESRI Token Authentication

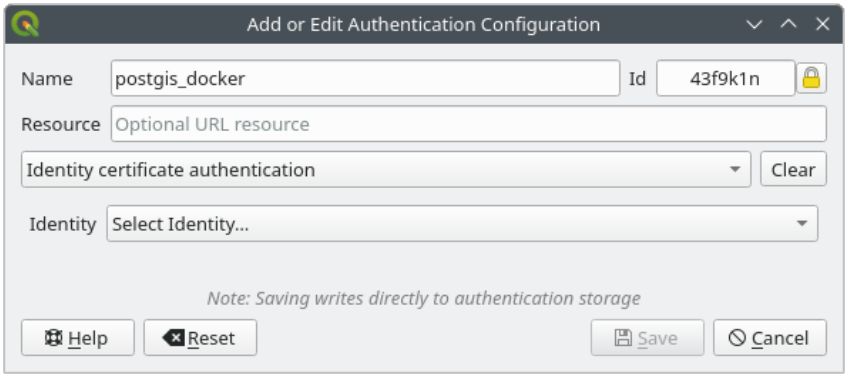
The ESRI token authentication method is used for *ArcGIS REST Servers* that require token-based authentication.



21.12 Fig. : ESRI Token authentication configs

Identity certificate authentication

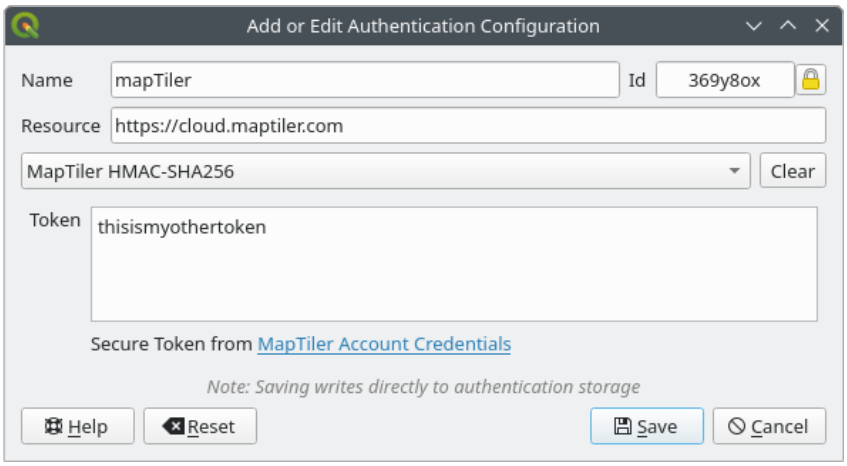
The Identity certificate authentication method allows you to connect using a client *identity certificate*.



21.13 Fig. : Identity-cert authentication configs

MapTiler HMAC-SHA256 Authentication

The MapTiler HMAC-SHA256 authentication method is used to connect to MapTiler services that require HMAC-SHA256 authorization.



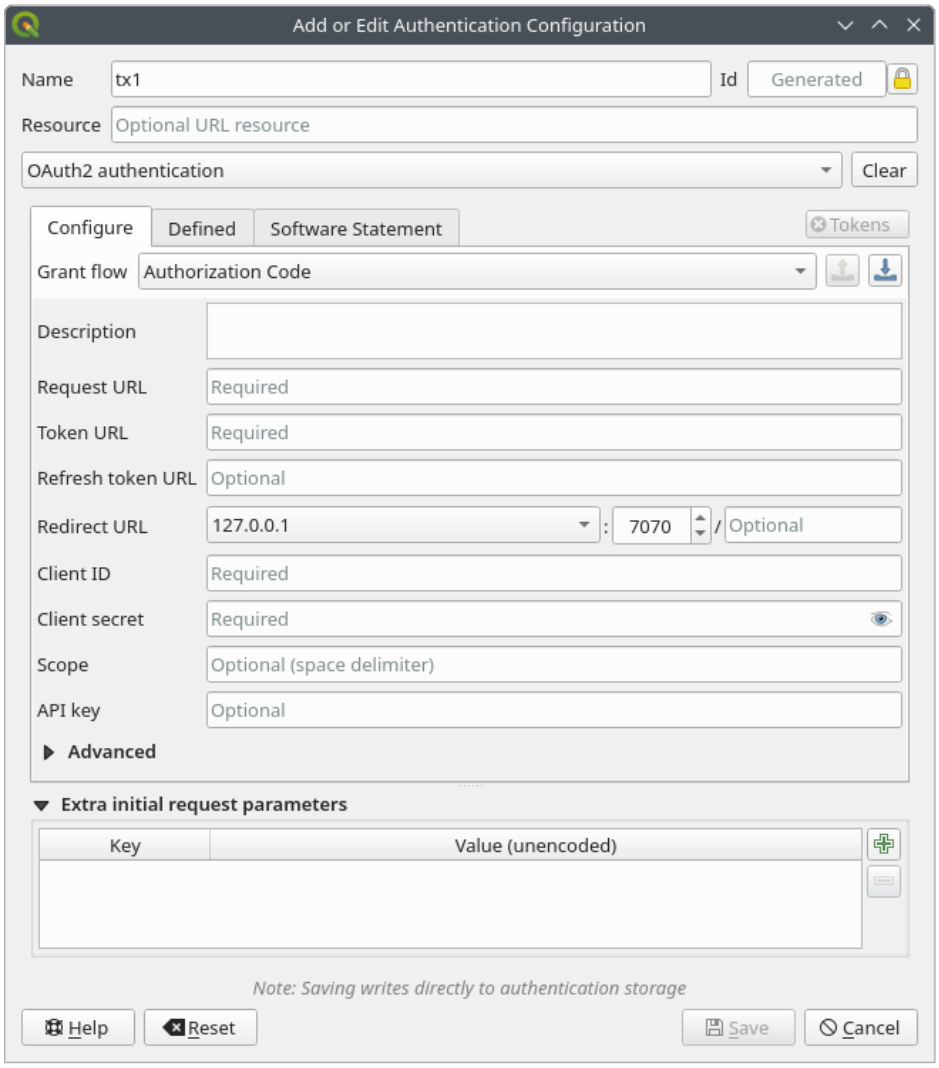
21.14 Fig. : MapTiler HmacSha256 authentication configs

OAuth2 Authentication

The OAuth2 authentication method is used to connect to services that require OAuth2 2.0 authorization, allowing secure access using client credential and token-based authentication.

It supports using `localhost` as the redirect host, providing compatibility with services that do not accept `http://127.0.0.1` redirects (for example, Microsoft SharePoint).

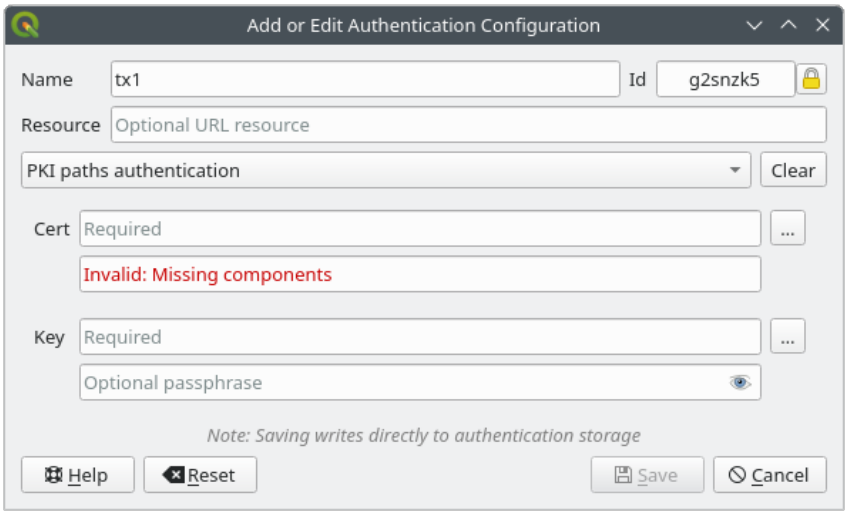
An advanced OAuth2 configuration option, *Extra token header(s)*, allows attaching additional tokens returned by the OAuth2 token endpoint as HTTP(S) request headers.



21.15 Fig. : OAuth2 authentication configs

PKI-Paths Authentication

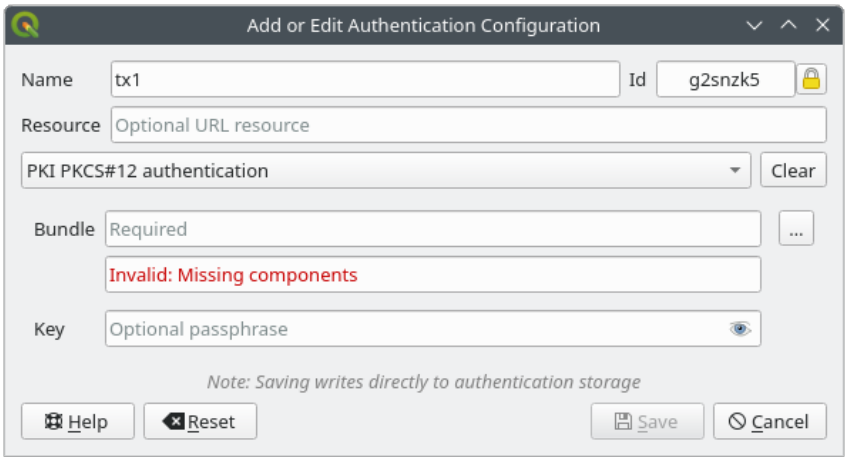
The PKI-Paths authentication method allows you to connect using separate certificate and key files stored on your system.



21.16 Fig. : PKI paths authentication configs

PKI-PKCS#12 Authentication

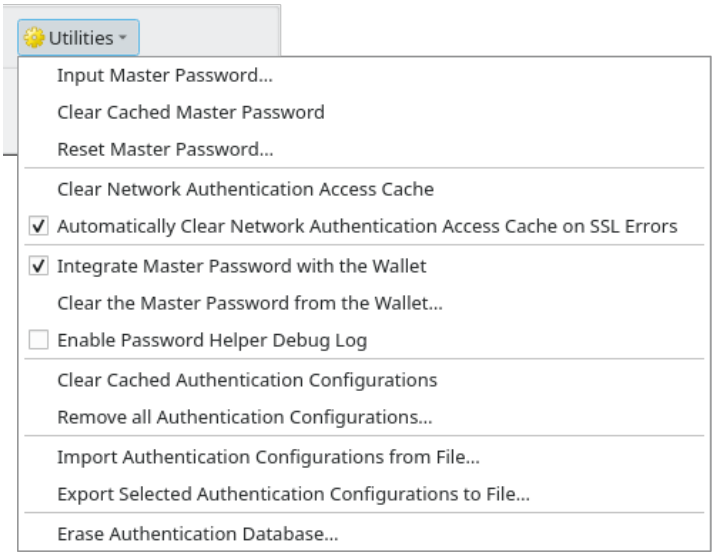
The PKI-PKCS#12 authentication method allows you to connect using a single bundle file containing both the certificate and private key, with an optional passphrase for the key.



21.17 Fig. : PKI PKCS#12 file paths authentication configs

21.1.6 Master Password and Auth Config Utilities

Under the Options menu (*Settings* ► *Options*) in the *Authentication* tab, there are several utility actions to manage the authentication database and configurations:



21.18 Fig. : Utilities menu

- *Input master password…*: opens the master password input dialog, independent of performing any authentication database command
- *Clear cached master password*: unsets the master password if it has been set
- *Reset master password…*: opens a dialog to change the master password (the current password must be known) and optionally back up the current database
- *Clear network authentication access cache*: clears the authentication cache of all connections
- *Automatically clear network authentication access cache on SSL errors*: the connection cache stores all authentication data for connections, also when the connection fails. If you change authentication configurations or certification authorities, you should clear the authentication cache or restart QGIS. When this option is checked, the authentication cache will be automatically cleared every time an SSL error occurs and you choose to abort the connection
- *Integrate master password with the Wallet*: adds the master password to your personal Wallet/Keyring
- *Clear the master password from the Wallet…*: deletes the master password from your Wallet/Keyring
- *Enable password helper debug log*: enables a debug tool that will contain all the log information of the authentication methods
- *Clear cached authentication configurations*: clears the internal lookup cache for configurations, used to speed up network connections. This does not clear QGIS's core network access manager's cache, which requires a relaunch of QGIS.
- *Remove all authentication configurations…*: clears the database of all configuration records, without removing other stored records.
- *Import authentication configurations from file…*: imports from an `.XML` file details for creating custom authentication configurations.
- *Export selected authentication configurations to file…*: exports the selected items to a possibly encrypted `.XML` file.

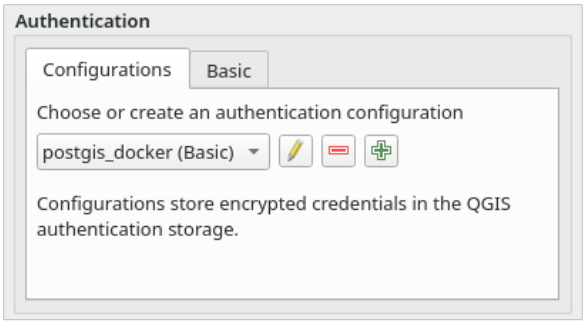
- *Erase authentication database…*: schedules a backup of the current database and complete rebuild of the database table structure. The actions are scheduled for a later time, to ensure that other operations, like project loading, do not interrupt the operation or cause errors due to a temporarily missing database.

21.1.7 Using authentication configurations

Typically, an authentication configuration is selected in a configuration dialog for a network services (such as WMS). However, the selector widget can be embedded anywhere authentication is needed or in non-core functionality, like in third-party PyQGIS or C++ plugins.

When using the selector, *No authentication* is displayed in the pop-up menu control when nothing is selected, when there are no configurations to choose from, or when a previously assigned configuration can no longer be found in the database. Use the drop-down menu to select an existing authentication configuration or press

to create a configuration you could use. More details at *Autentifikacija*.



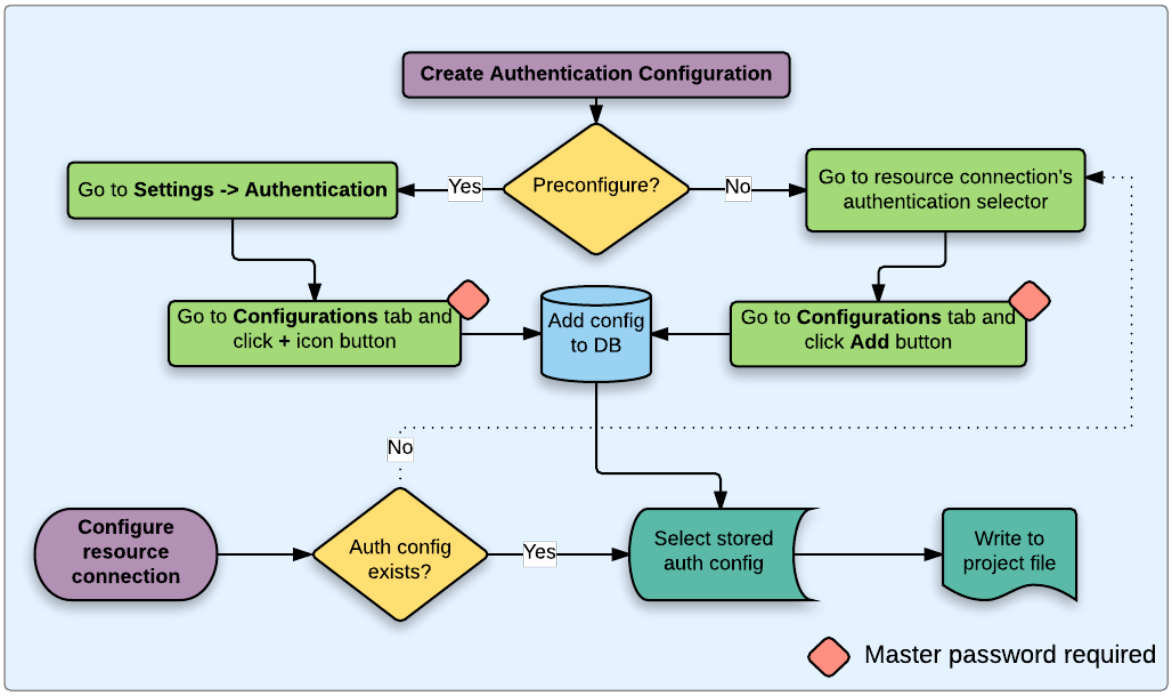
21.19 Fig. : Authentication configuration selector with selected config

21.1.8 Python bindings

All classes and public functions have sip bindings, except QgsAuthCrypto, since management of the master password hashing and auth database encryption should be handled by the main app, and not via Python.

Once the master password is entered, the API is open to access authentication configs in the authentication database, similar to how Firefox works. However, no wall against PyQGIS access has been defined. This may lead to issues where a user downloads/installs a malicious PyQGIS plugin or standalone app that gains access to authentication credentials.

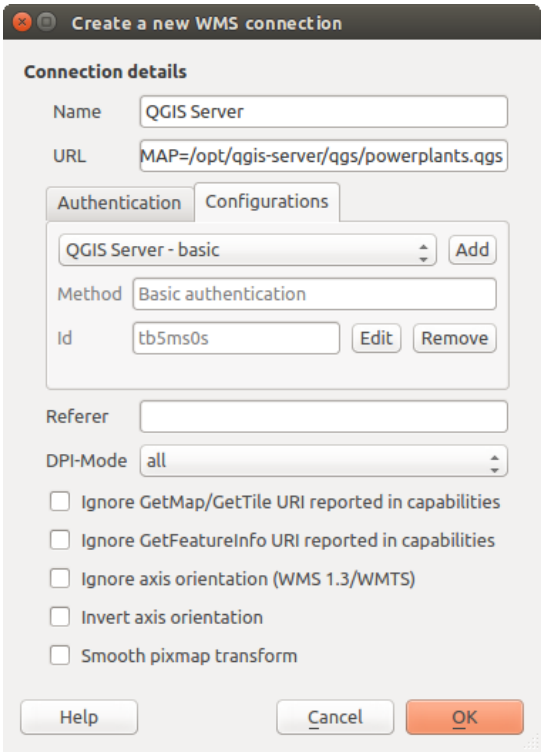
21.2 User Authentication Workflows



21.20 Fig. : Generic user workflow

21.2.1 HTTP(S) authentication

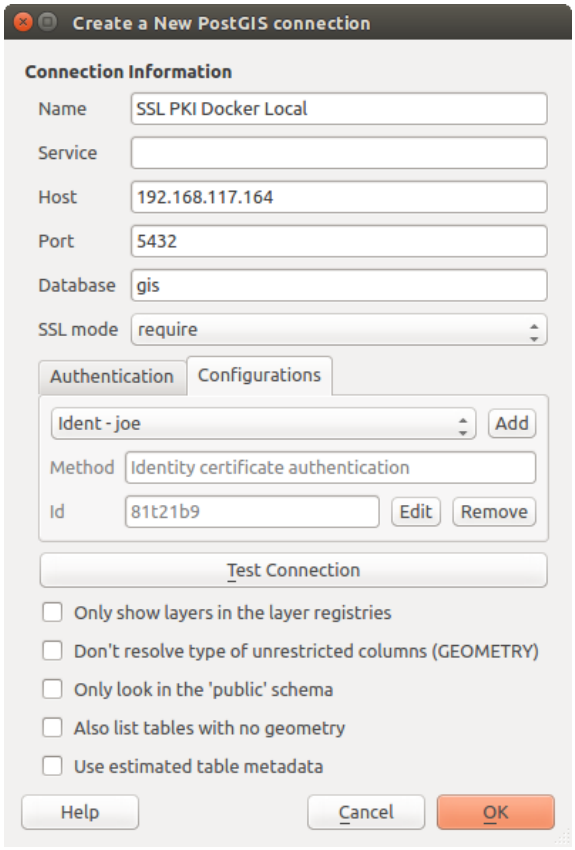
One of the most common resource connections is via HTTP(S), e.g. web mapping servers, and authentication method plugins often work for these types of connections. Method plugins have access to the HTTP request object and can manipulate both the request as well as its headers. This allows for many forms of internet-based authentication. When connecting via HTTP(S) using the standard username/password authentication method will attempt HTTP BASIC authentication upon connection.



21.21 Fig. : Configuring a WMS connection for HTTP BASIC

21.2.2 Database authentication

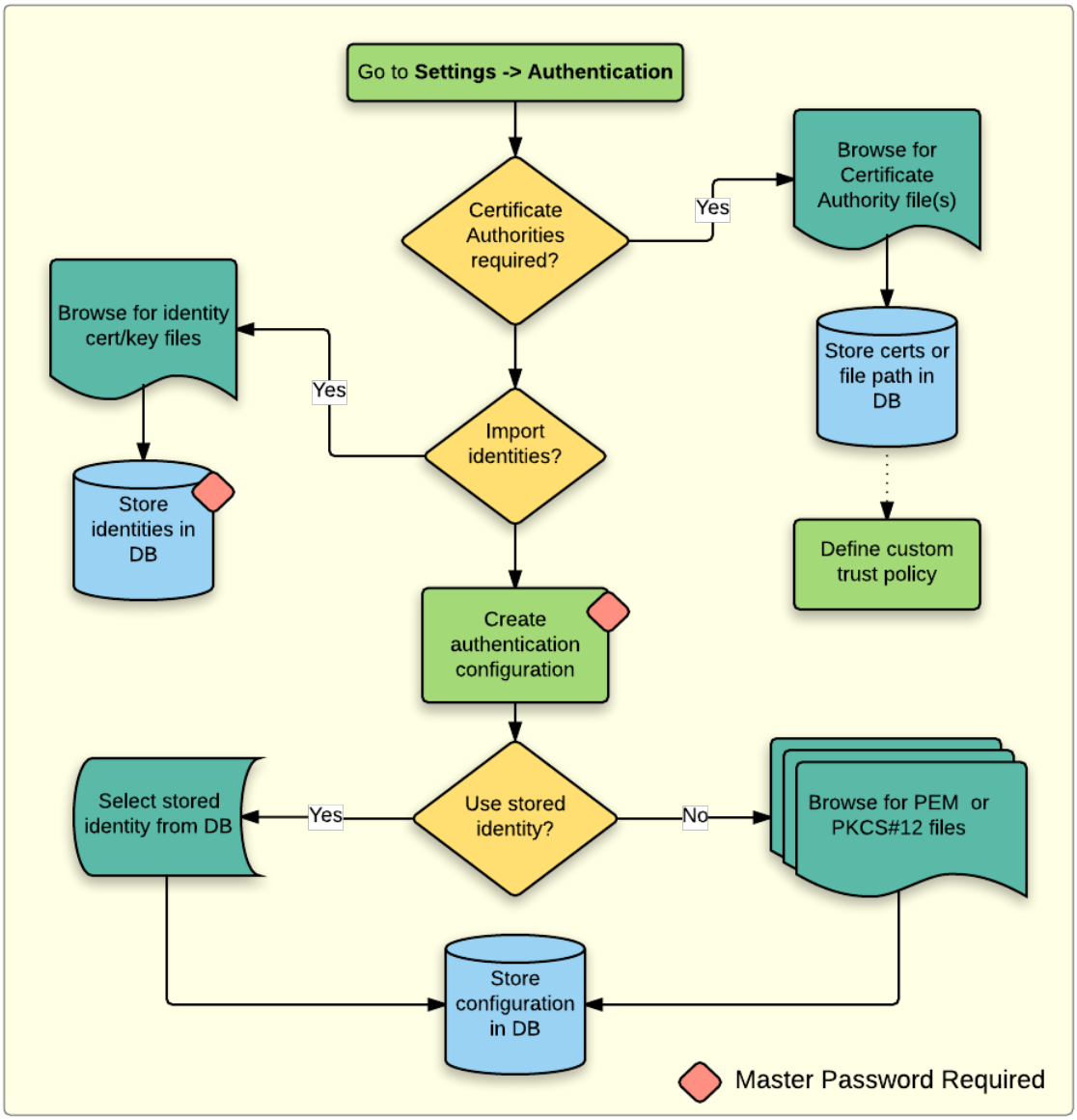
Connections to database resources are generally stored as `key=value` pairs, which will expose usernames and (optionally) passwords, if *not* using an authentication configuration. When configuring with the auth system, the `key=value` will be an abstracted representation of the credentials, e.g. `authfg=81t21b9`.



21.22 Fig. : Configuring a Postgres SSL-with-PKI connection

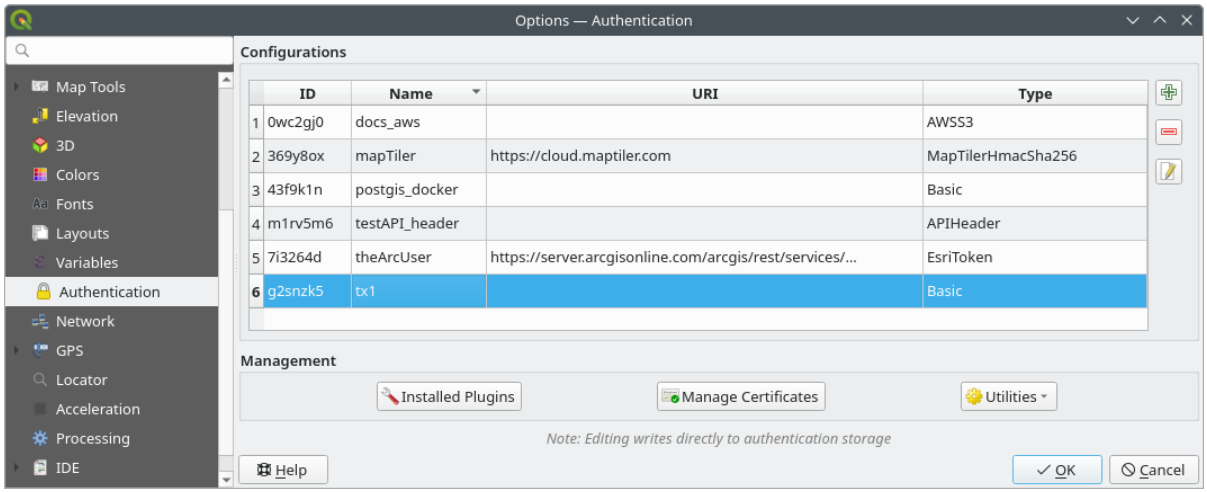
21.2.3 PKI authentication

When configuring PKI components within the authentication system, you have the option of importing components into the database or referencing component files stored on your filesystem. The latter may be useful if such components change frequently, or where the components will be replaced by a system administrator. In either instance you will need to store any passphrase needed to access private keys within the database.



21.23 Fig. : PKI configuration workflow

All PKI components can be managed in separate editors within the **Certificate Manager**, which can be accessed in the *Authentication* tab in QGIS *Options* dialog (*Settings* ► *Options*) by clicking the *Manage Certificates* button.



21.24 Fig. : Opening the Certificate Manager

In the *Certificate Manager*, there are editors for **Identities**, **Servers** and **Authorities**. Each of these are contained in their own tabs, and are described below in the order they are encountered in the workflow chart above. The tab order is relative to frequently accessed editors once you are accustomed to the workflow.

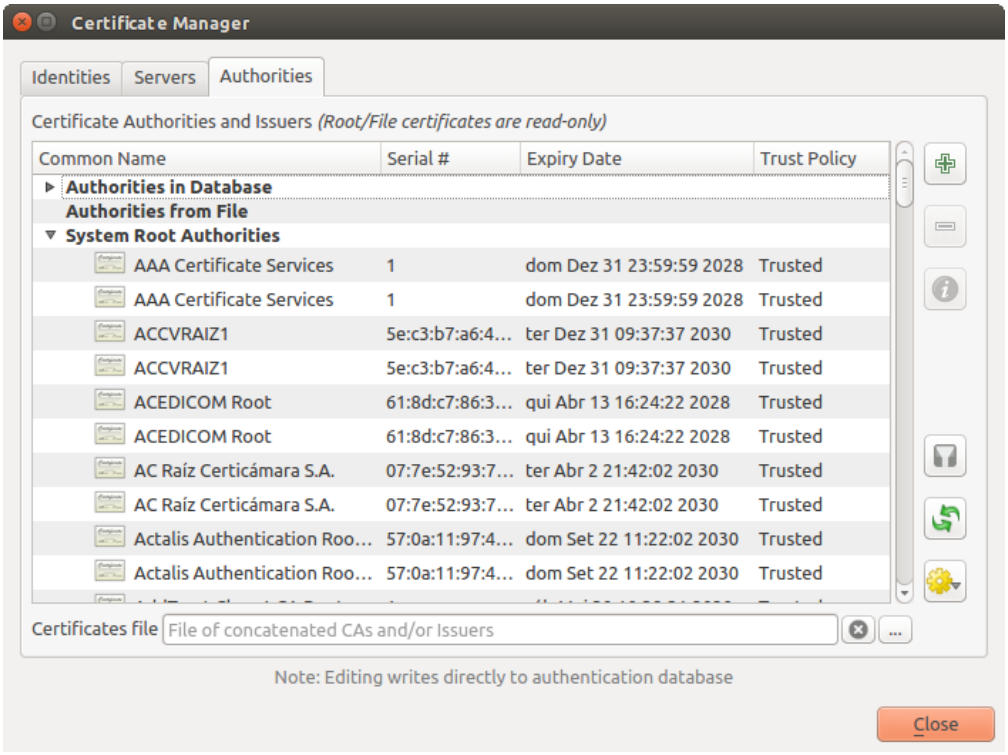
Pastaba

Because all authentication system edits write immediately to the authentication database, there is no need to click the *Options* dialog *OK* button for any changes to be saved. This is unlike other settings in the Options dialog.

Authorities

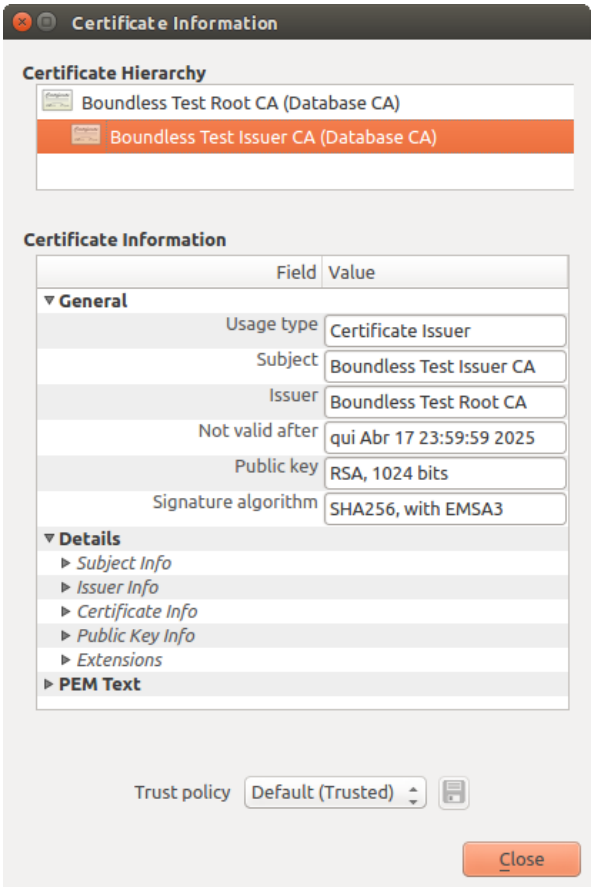
You can manage available Certificate Authorities (CAs) from the **Authorities** tab in the **Certificate manager** from the **Authentication** tab of the QGIS **Options** dialog.

As referenced in the workflow chart above, the first step is to import or reference a file of CAs. This step is optional, and may be unnecessary if your PKI trust chain originates from root CAs already installed in your operating system (OS), such as a certificate from a commercial certificate vendor. If your authenticating root CA is not in the OS's trusted root CAs, it will need to be imported or have its file system path referenced. (Contact your system administrator if unsure.)



21.25 Fig. : Authorities editor

By default, the root CAs from your OS are available; however, their trust settings are not inherited. You should review the certificate trust policy settings, especially if your OS root CAs have had their policies adjusted. Any certificate that is expired will be set to untrusted and will not be used in secure server connections, unless you specifically override its trust policy. To see the QGIS-discoverable trust chain for any certificate, select it and click the Show information for certificate.



21.26 Fig. : Certificate info dialog

You can edit the *Trust policy* for any selected certificate within the chain. Any change in trust policy to a selected certificate will not be saved to the database unless the Save certificate trust policy change to database button is clicked *per* selected certification. Closing the dialog will **not** apply the policy changes.

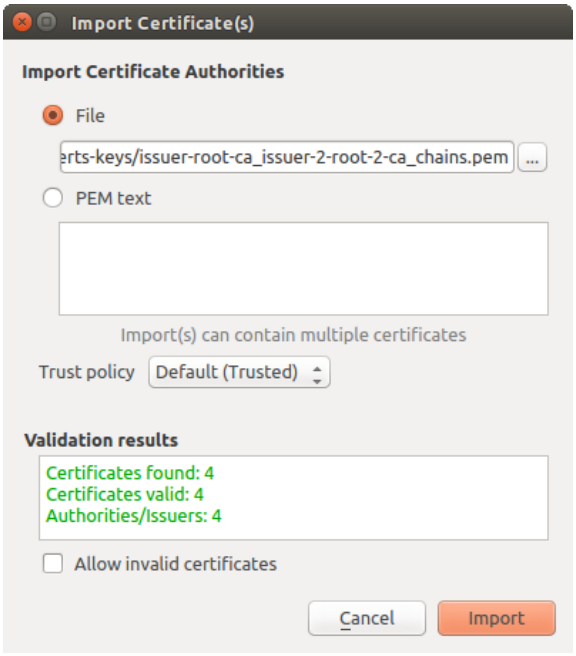
You can review the filtered CAs, both intermediate and root certificates, that will be trusted for secure connections or change the default trust policy by clicking the **Options** button.

Įspėjimas

Changing the default trust policy may result in problems with secure connections.

You can import CAs or save a file system path from a file that contains multiple CAs, or import individual CAs. The standard PEM format for files that contain multiple CA chain certifications has the root cert at the bottom of the file and all subsequently signed child certificates above, towards the beginning of the file.

The CA certificate import dialog will find all CA certificates within the file, regardless of order, and also offers the option to import certificates that are considered invalid (in case you want to override their trust policy). You can override the trust policy upon import, or do so later within the **Authorities** editor.



21.27 Fig. : Import certificates dialog

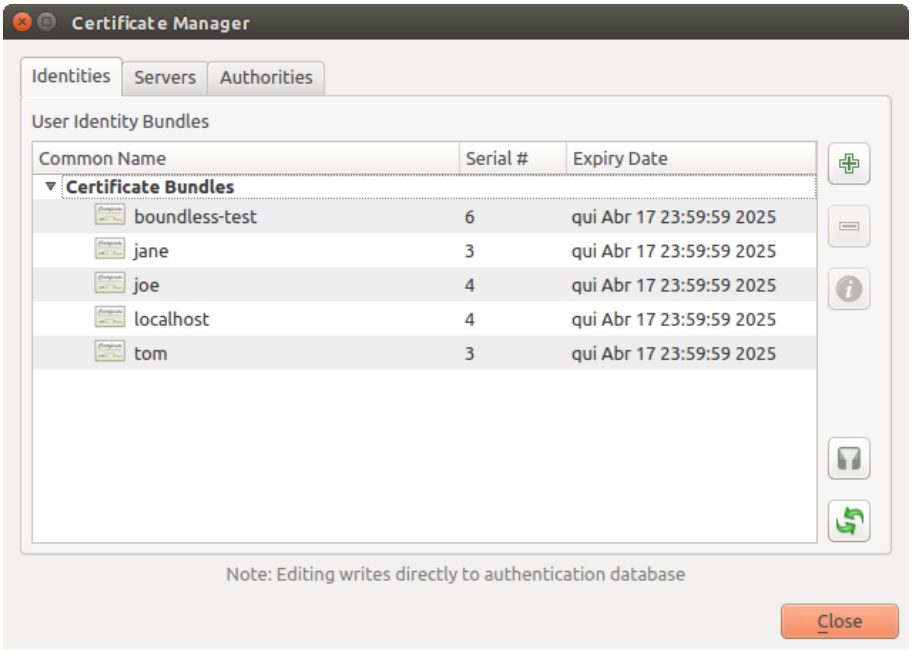
Pastaba

If you are pasting certificate information into the *PEM text* field, note that encrypted certificates are not supported.

Identities

You can manage available client identity bundles from the *Identities* tab in the *Certificate manager* from the **Authentication** tab of the QGIS **Options** dialog. An identity is what authenticates you against a PKI-enabled service and usually consists of a client certificate and private key, either as separate files or combined into a single „bundled“ file. The bundle or private key is often passphrase-protected.

Once you have any Certificate Authorities (CAs) imported you can optionally import any identity bundles into the authentication database. If you do not wish to store the identities, you can reference their component file system paths within an individual authentication configuration.

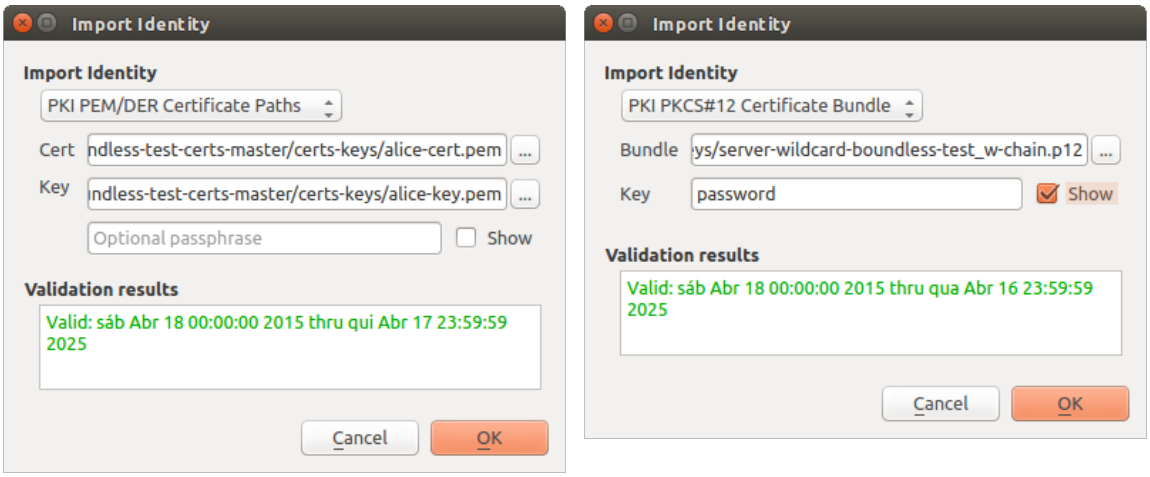


21.28 Fig. : Identities editor

When importing an identity bundle, it can be passphrase-protected or unprotected, and can contain CA certificates forming a trust chain. Trust chain certifications will not be imported here; they can be added separately under the *Authorities* tab.

Upon import the bundle's certificate and private key will be stored in the database, with the key's storage encrypted using the QGIS master password. Subsequent usage of the stored bundle from the database will only require input of the master password.

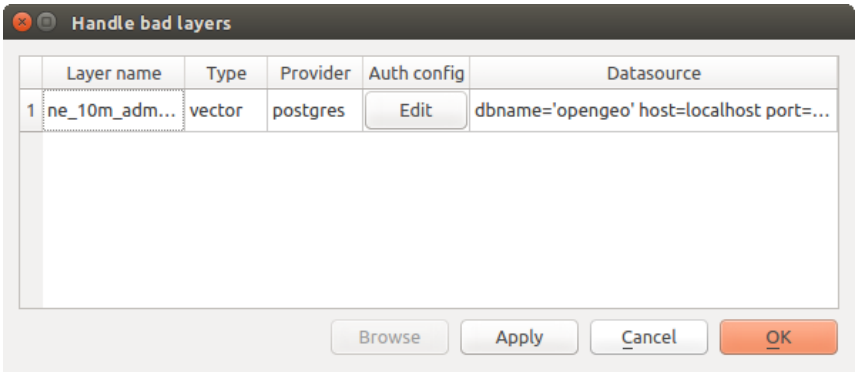
Personal identity bundles consisting of PEM/DER (.pem/.der) and PKCS#12 (.p12/.pfx) components are supported. If a key or bundle is passphrase-protected, the password will be required to validate the component prior to import. Likewise, if the client certificate in the bundle is invalid (for example, its effective date has not yet started or has elapsed) the bundle can not be imported.



21.29 Fig. : PEM/DER and PKCS#12 identities import

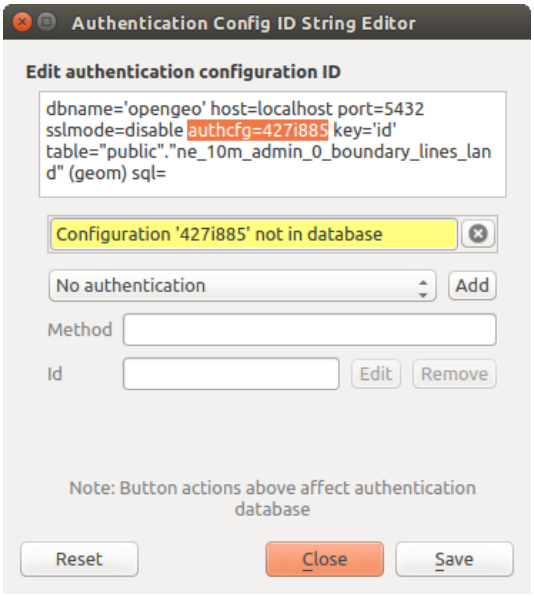
21.2.4 Handling bad layers

Occasionally, the authentication configuration ID that is saved with a project file is no longer valid, possibly because the current authentication database is different than when the project was last saved, or due to a credentials mismatch. In such cases the *Handle bad layers* dialog will be presented upon QGIS launch.



21.30 Fig. : Handle bad layers with authentication

If a data source is found to have an authentication configuration ID associated with it, you will be able to edit it. Doing so will automatically edit the data source string, much in the same way as opening the project file in a text editor and editing the string.

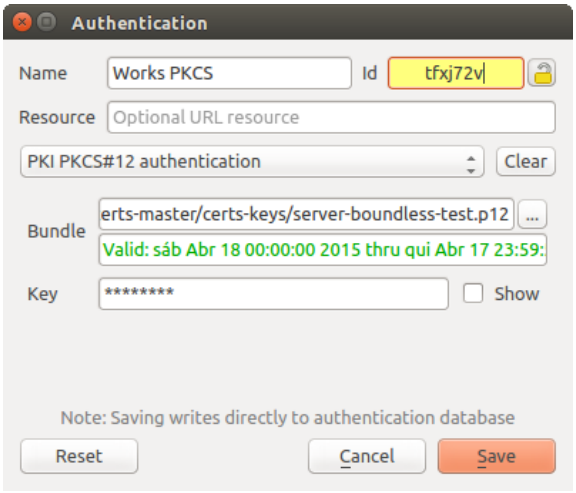


21.31 Fig. : Edit bad layer's authentication config ID

21.2.5 Changing authentication config ID

Occasionally, you will need to change the authentication configuration ID that is associated with accessing a resource. There are instances where this is useful:

- **Resource auth config ID is no longer valid**: This can occur when you have switched auth databases add need to *align* a new configuration to the ID already associated with a resource.
- **Shared project files**: If you intended to share projects between users, e.g. via a shared file server, you can *predefine* a 7-character (containing **a-z** and/or **0-9**) that is associated with the resource. Then, individual users change the ID of an authentication configuration that is specific to their credentials of the resource. When the project is opened, the ID is found in the authentication database, but the credentials are different per user.



21.32 Fig. : Changing a layer's authentication config ID (unlocked yellow text field)

Įspėjimas

Changing the auth config ID is considered an advanced operation and should only be done with full knowledge as to why it is necessary. This is why there is a lock button that needs clicked, to unlock the ID's text field prior to editing the ID.

21.2.6 QGIS Server support

When using a project file, with layers that have authentication configurations, as a basis for a map in QGIS Server, there are a couple of additional setup steps necessary for QGIS to load the resources:

- Authentication database needs to be available
- Authentication database's master password needs to be available

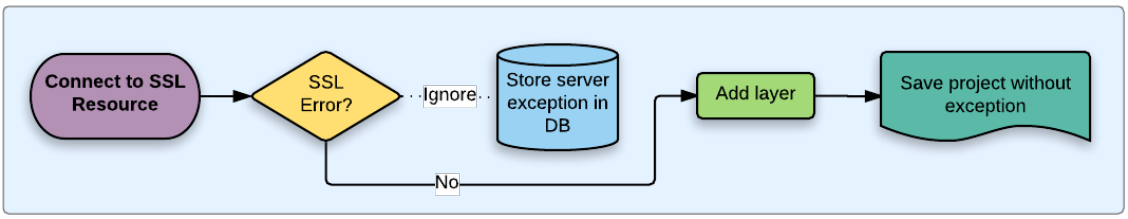
When instantiating the authentication system, Server will create or use `qgis-auth.db` file in the active *user profile*, or the directory defined by the `QGIS_AUTH_DB_DIR_PATH` environment variable. It may be that the Server's user has no HOME directory, in which case, use the environment variable to define a directory that the Server's user has read/write permissions and is not located within the web-accessible directories.

To pass the master password to Server, write it to the first line of file at a path on the file system readable by the Server processes user and defined using the `QGIS_AUTH_PASSWORD_FILE` environment variable. Ensure to limit the file as only readable by the Server's process user and to not store the file within web-accessible directories.

Pastaba

`QGIS_AUTH_PASSWORD_FILE` variable will be removed from the Server environment immediately after accessing.

21.2.7 SSL server exceptions



21.33 Fig. : SSL server exception

You can manage SSL server configurations and exceptions from the **Servers** tab in the **Authentication** section of the QGIS **Options** dialog.

Sometimes, when connecting to an SSL server, there are errors with the SSL „handshake“ or the server's certificate. You can ignore those errors or create an SSL server configuration as an exception. This is similar to how web browsers allow you to override SSL errors, but with more granular control.

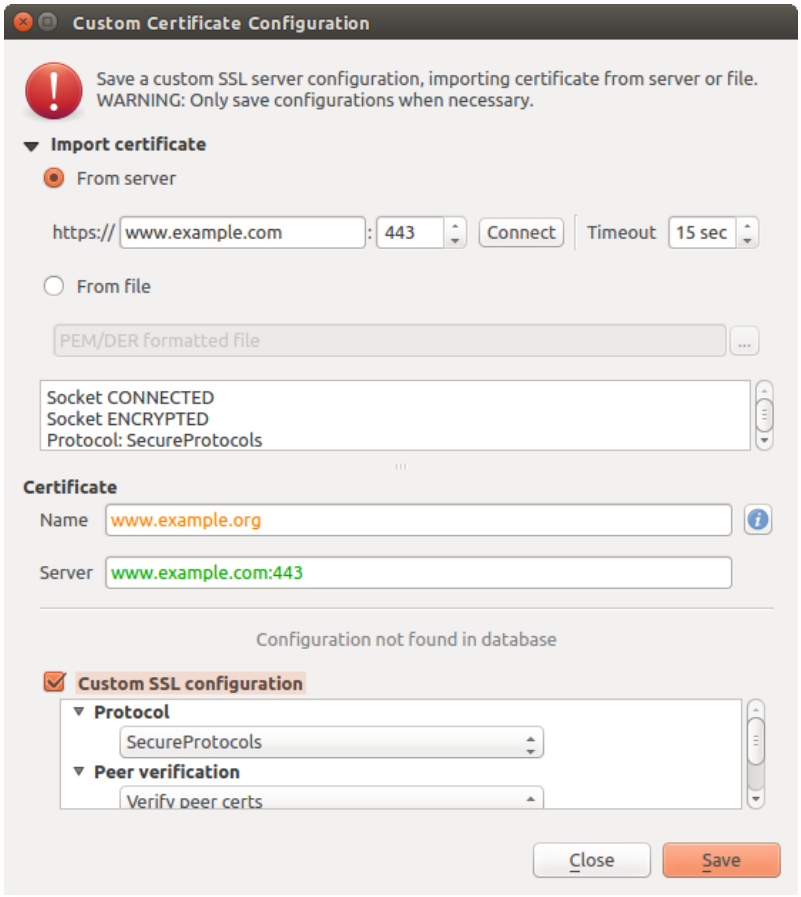
Įspėjimas

You should not create an SSL server configuration unless you have complete knowledge of the entire SSL setup between the server and client. Instead, report the issue to the server administrator.

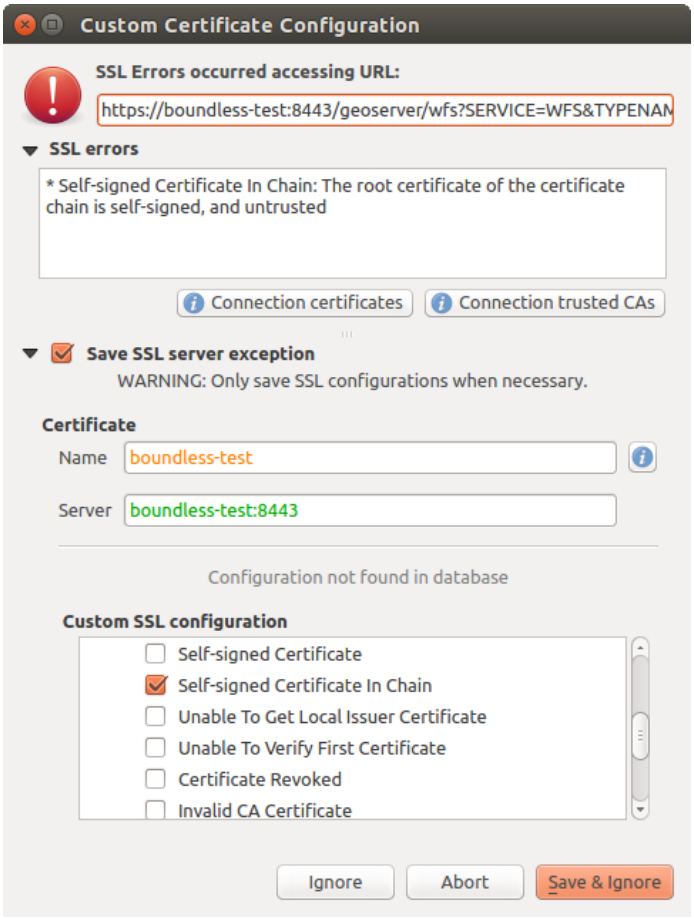
Pastaba

Some PKI setups use a completely different CA trust chain to validate client identities than the chain used to validate the SSL server certificate. In such circumstances, any configuration created for the connecting server will not necessarily fix an issue with the validation of your client identity, and only your client identity's issuer or server administrator can fix the issue.

You can pre-configure an SSL server configuration by clicking the button. Alternatively, you can add a configuration when an SSL error occurs during a connection and you are presented with an **SSL Error** dialog (where the error can be ignored temporarily or saved to the database and ignored):

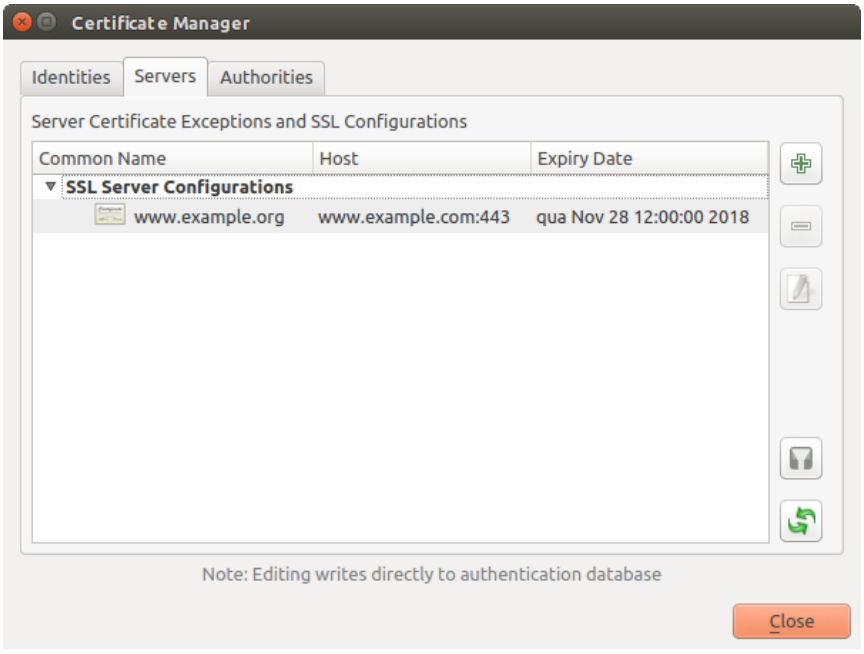


21.34 Fig. : Manually adding configuration

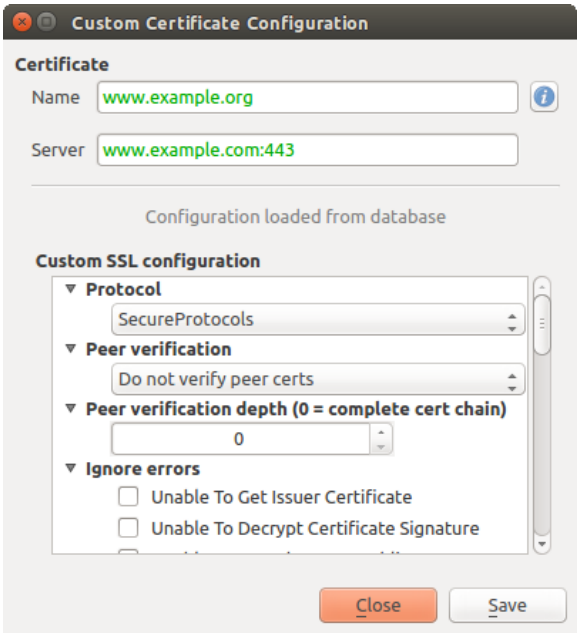


21.35 Fig. : Adding configuration during SSL error

Once an SSL configuration is saved to the database, it can be edited or deleted.



21.36 Fig. : Existing SSL configuration



21.37 Fig. : Editing an existing SSL configuration

If you want to pre-configure an SSL configuration and the import dialog is not working for your server's connection, you can manually trigger a connection via the **Python Console** by running the following code (replace `https://bugreports.qt-project.org` with the URL of your server):

```
from qgis.PyQt.QtNetwork import QNetworkRequest
from qgis.PyQt.QtCore import QUrl
from qgis.core import QgsNetworkAccessManager

req = QNetworkRequest(QUrl('https://bugreports.qt-project.org'))
reply = QgsNetworkAccessManager.instance().get(req)
```

This will open an SSL error dialog if any errors occur, where you can choose to save the configuration to the database.

SKYRIUS 22

GRASS GIS Integration

GRASS integration provides access to GRASS GIS databases and functionalities (see GRASS-PROJECT in *Literature and Web References*). The integration consists of two parts: provider and plugin. The provider allows to browse, manage and visualize GRASS raster and vector layers. The plugin can be used to create new GRASS locations and mapsets, change GRASS region, create and edit vector layers and analyze GRASS 2-D and 3-D data with more than 400 GRASS modules. In this section, we'll introduce the provider and plugin functionalities and give some examples of managing and working with GRASS data.

The provider supports GRASS version 6 and 7, the plugin supports GRASS 6 and 7 (starting from QGIS 2.12). QGIS distribution may contain provider/plugin for either GRASS 6 or GRASS 7 or for both versions at the same time (binaries have different file names). Only one version of the provider/plugin may be loaded on runtime however.

22.1 Demo dataset

As an example, we will use the QGIS Alaska dataset (see section *Pavyzdinių duomenų atsisiuntimas*). It includes a small sample GRASS `LOCATION` with three vector layers and one raster elevation map. Create a new folder called `grassdata`, download the QGIS ‚Alaska' dataset `qgis_sample_data.zip` from https://qgis.org/downloads-list/#data and unzip the file into `grassdata`.

More sample GRASS `LOCATIONs` are available at the GRASS website.

22.2 Loading GRASS raster and vector layers

If the provider is loaded in QGIS, the location item with GRASS icon is added in the browser tree under each folder item which contains GRASS location. Go to the folder `grassdata` and expand location `alaska` and mapset `demo`.

You can load GRASS raster and vector layers like any other layer from the browser either by double click on layer item or by dragging and dropping to map canvas or legend.

Patarimas

GRASS Data Loading

If you don't see GRASS location item, verify in *Help* ► *About* ► *Providers* if GRASS vector provider is loaded.

22.3 Importing data into a GRASS LOCATION via drag and drop

This section gives an example of how to import raster and vector data into a GRASS mapset.

1. In QGIS browser navigate to the mapset you want to import data into.
2. In QGIS browser find a layer you want to import to GRASS, note that you can open another instance of the browser (*Browser Panel (2)*) if source data are too far from the mapset in the tree.
3. Drag a layer and drop it on the target mapset. The import may take some time for larger layers, you will see animated icon in front of new layer item until the import finishes.

When raster data are in different CRS, they can be reprojected using an *Approximate* (fast) or *Exact* (precise) transformation. If a link to the source raster is created (using `r.external`), the source data are in the same CRS and the format is known to GDAL, the source data CRS will be used. You can set these options in the *Browser* tab in *GRASS Options*.

If a source raster has more bands, a new GRASS map is created for each layer with **.<band number>** suffix and group of all maps with icon is created. External rasters have a different icon .

22.4 Managing GRASS data in QGIS Browser

- Copying maps: GRASS maps may be copied between mapsets within the same location using drag and drop.
- Deleting maps: Right click on a GRASS map and select *Delete* from context menu.
- Renaming maps: Right click on a GRASS map and select *Rename* from context menu.

22.5 GRASS Options

GRASS options may be set in *GRASS Options* dialog, which can be opened by right clicking on the location or mapset item in the browser and then choosing *GRASS Options*.

22.6 Starting the GRASS plugin

To use GRASS functionalities in QGIS, you must select and load the GRASS plugin using the Plugin Manager. To do this, go to the menu *Plugins* ► *Manage and Install Plugins…*, select *GRASS* and click *OK*.

The following main features are provided with the GRASS menu (*Plugins* ► *GRASS*) when you start the GRASS plugin:

- Open Mapset
- New Mapset
- Close Mapset
- Open GRASS Tools
- Display Current GRASS Region

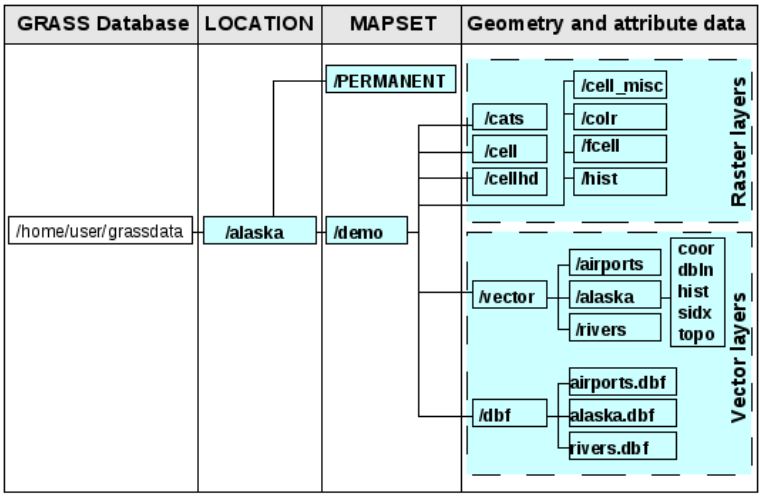
- GRASS Options

22.7 Opening GRASS mapset

A GRASS mapset must be opened to get access to GRASS Tools in the plugin (the tools are disabled if no mapset is open). You can open a mapset from the browser: right click on mapset item and then choose *Open mapset* from context menu.

22.8 GRASS LOCATION and MAPSET

GRASS data are stored in a directory referred to as GISDBASE. This directory, often called `grassdata`, must be created before you start working with the GRASS plugin in QGIS. Within this directory, the GRASS GIS data are organized by projects stored in subdirectories called `LOCATIONs`. Each `LOCATION` is defined by its coordinate system, map projection and geographical boundaries. Each `LOCATION` can have several `MAPSETs` (subdirectories of the `LOCATION`) that are used to subdivide the project into different topics or sub-regions, or as workspaces for individual team members (see Neteler & Mitasova 2008 in *Literature and Web References*). In order to analyze vector and raster layers with GRASS modules, you generally have to import them into a GRASS `LOCATION`. (This is not strictly true – with the GRASS modules `r.external` and `v.external` you can create read-only links to external GDAL-supported datasets without importing them. This is not the usual way for beginners to work with GRASS, therefore this functionality will not be described here.)



22.1 Fig. : GRASS data in the alaska LOCATION

22.9 Importing data into a GRASS LOCATION

See section *Importing data into a GRASS LOCATION via drag and drop* to find how data can be easily imported by dragging and dropping in the browser.

This section gives an example of how to import raster and vector data into the ‚alaska' GRASS `LOCATION` provided by the QGIS ‚Alaska' dataset in traditional way, using standard GRASS modules. Therefore, we use the landcover raster map `landcover.img` and the vector GML file `lakes.gml` from the QGIS ‚Alaska' dataset (see *Pavyzdinių duomenų atsisiuntimas*).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. In the GRASS toolbar, click the Open MAPSET icon to bring up the *MAPSET* wizard.

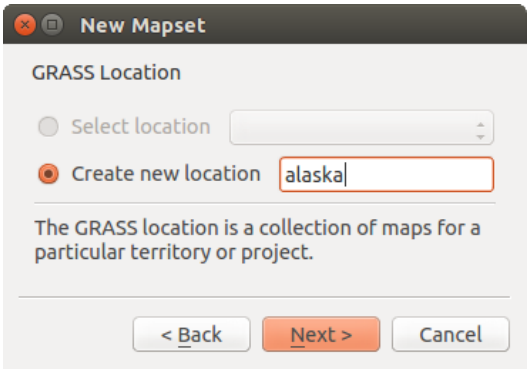
3. Select as GRASS database the folder `grassdata` in the QGIS Alaska dataset, as `LOCATION` ‚alaska', as `MAPSET` ‚demo' and click *OK*.
4. Now click the Open GRASS tools icon. The GRASS Toolbox (see section *The GRASS Toolbox*) dialog appears.
5. To import the raster map `landcover.img`, click the module `r.in.gdal` in the *Modules Tree* tab. This GRASS module allows you to import GDAL-supported raster files into a GRASS `LOCATION`. The module dialog for `r.in.gdal` appears.
6. Browse to the folder `raster` in the QGIS ‚Alaska' dataset and select the file `landcover.img`.
7. As raster output name, define `landcover_grass` and click *Run*. In the *Output* tab, you see the currently running GRASS command `r.in.gdal -o input=/path/to/landcover.img output=landcover_grass`.
8. When it says **Successfully finished**, click *View Output*. The `landcover_grass` raster layer is now imported into GRASS and will be visualized in the QGIS canvas.
9. To import the vector GML file `lakes.gml`, click the module `v.in.ogr` in the *Modules Tree* tab. This GRASS module allows you to import OGR-supported vector files into a GRASS `LOCATION`. The module dialog for `v.in.ogr` appears.
10. Browse to the folder `gml` in the QGIS ‚Alaska' dataset and select the file `lakes.gml` as OGR file.
11. As vector output name, define `lakes_grass` and click *Run*. You don't have to care about the other options in this example. In the *Output* tab you see the currently running GRASS command `v.in.ogr -o dsn=/path/to/lakes.gml output=lakes\_grass`.
12. When it says **Succesfully finished**, click *View Output*. The `lakes_grass` vector layer is now imported into GRASS and will be visualized in the QGIS canvas.

22.9.1 Creating a new GRASS LOCATION

As an example, here is the sample GRASS `LOCATION alaska`, which is projected in the Albers Equal Area projection using feet as units. This sample GRASS `LOCATION alaska` will be used for all examples and exercises in the following GRASS-related sections. It is useful to download and install the dataset on your computer (see *Pavyzdinių duomenų atsisiuntimas*).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. Visualize the `alaska.shp` shapefile (see section *Sluoksnio įkėlimas iš failo*) from the QGIS Alaska dataset (see *Pavyzdinių duomenų atsisiuntimas*).
3. In the GRASS toolbar, click on the New mapset icon to bring up the *MAPSET* wizard.
4. Select an existing GRASS database (GISDBASE) folder `grassdata`, or create one for the new `LOCATION` using a file manager on your computer. Then click *Next*.
5. We can use this wizard to create a new `MAPSET` within an existing `LOCATION` (see section *Adding a new MAPSET*) or to create a new `LOCATION` altogether. Select *Create new location* (see Fig. 22.2).
6. Enter a name for the `LOCATION` – we used ‚alaska' – and click *Next*.
7. Define the projection by clicking on the radio button *Projection* to enable the projection list.
8. We are using Albers Equal Area Alaska (feet) projection. Since we happen to know that it is represented by the EPSG ID 2964, we enter it in the search box. (Note: If you want to repeat this process for another `LOCATION` and projection and haven't memorized the EPSG ID, click on the CRS Status icon in the lower right-hand corner of the status bar (see section *Darbas su projekcijomis*)).
9. In *Filter*, insert 2964 to select the projection.
10. Click *Next*.

11. To define the default region, we have to enter the LOCATION bounds in the north, south, east, and west directions. Here, we simply click on the button *Set Current QGIS Extent*, to apply the extent of the loaded layer `alaska.shp` as the GRASS default region extent.
12. Click *Next*.
13. We also need to define a MAPSET within our new LOCATION (this is necessary when creating a new LOCATION). You can name it whatever you like - we used ‚demo'. GRASS automatically creates a special MAPSET called PERMANENT, designed to store the core data for the project, its default spatial extent and coordinate system definitions (see Neteler & Mitasova 2008 in *Literature and Web References*).
14. Check out the summary to make sure it's correct and click *Finish*.
15. The new LOCATION, ‚alaska', and two MAPSETs, ‚demo' and ‚PERMANENT', are created. The currently opened working set is ‚demo', as you defined.
16. Notice that some of the tools in the GRASS toolbar that were disabled are now enabled.



22.2 Fig. : Creating a new GRASS LOCATION or a new MAPSET in QGIS

If that seemed like a lot of steps, it's really not all that bad and a very quick way to create a LOCATION. The LOCATION ‚alaska' is now ready for data import (see section *Importing data into a GRASS LOCATION*). You can also use the already-existing vector and raster data in the sample GRASS LOCATION ‚alaska', included in the QGIS ‚Alaska' dataset *Pavyzdinių duomenų atsisiuntimas*, and move on to section *The GRASS vector data model*.

22.9.2 Adding a new MAPSET

A user has write access only to a GRASS MAPSET which he or she created. This means that besides access to your own MAPSET, you can read maps in other users' MAPSETs (and they can read yours), but you can modify or remove only the maps in your own MAPSET.

All MAPSETs include a WIND file that stores the current boundary coordinate values and the currently selected raster resolution (see Neteler & Mitasova 2008 in *Literature and Web References*, and section *The GRASS region tool*).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. In the GRASS toolbar, click on the New mapset icon to bring up the *MAPSET* wizard.
3. Select the GRASS database (GISDBASE) folder `grassdata` with the LOCATION ‚alaska', where we want to add a further MAPSET called ‚test'.
4. Click *Next*.
5. We can use this wizard to create a new MAPSET within an existing LOCATION or to create a new LOCATION altogether. Click on the radio button *Select location* (see Fig. 22.2) and click *Next*.
6. Enter the name `test` for the new MAPSET. Below in the wizard, you see a list of existing MAPSETs and corresponding owners.
7. Click *Next*, check out the summary to make sure it's all correct and click *Finish*.

22.10 The GRASS vector data model

It is important to understand the GRASS vector data model prior to digitizing. In general, GRASS uses a topological vector model. This means that areas are not represented as closed polygons, but by one or more boundaries. A boundary between two adjacent areas is digitized only once, and it is shared by both areas. Boundaries must be connected and closed without gaps. An area is identified (and labelled) by the **centroid** of the area.

Besides boundaries and centroids, a vector map can also contain points and lines. All these geometry elements can be mixed in one vector and will be represented in different so-called ‚layers' inside one GRASS vector map. So in GRASS, a layer is not a vector or raster map but a level inside a vector layer. This is important to distinguish carefully. (Although it is possible to mix geometry elements, it is unusual and, even in GRASS, only used in special cases such as vector network analysis. Normally, you should prefer to store different geometry elements in different layers.)

It is possible to store several ‚layers' in one vector dataset. For example, fields, forests and lakes can be stored in one vector. An adjacent forest and lake can share the same boundary, but they have separate attribute tables. It is also possible to attach attributes to boundaries. An example might be the case where the boundary between a lake and a forest is a road, so it can have a different attribute table.

The ‚layer' of the feature is defined by the ‚layer' inside GRASS. ‚Layer' is the number which defines if there is more than one layer inside the dataset (e.g., if the geometry is forest or lake). For now, it can be only a number. In the future, GRASS will also support names as fields in the user interface.

Attributes can be stored inside the GRASS `LOCATION` as dBase, SQLite3 or in external database tables, for example, PostgreSQL, MySQL, Oracle, etc.

Attributes in database tables are linked to geometry elements using a ‚category' value.

‚Category' (key, ID) is an integer attached to geometry primitives, and it is used as the link to one key column in the database table.

Patarimas

Learning the GRASS Vector Model

The best way to learn the GRASS vector model and its capabilities is to download one of the many GRASS tutorials where the vector model is described more deeply. See https://grass.osgeo.org/learn/manuals/ for more information, books and tutorials in several languages.

22.11 Creating a new GRASS vector layer

To create a new GRASS vector layer, select one of following items from mapset context menu in the browser:

- New Point Layer
- New Line Layer
- New Polygon Layer

and enter a name in the dialog. A new vector map will be created and layer will be added to canvas and editing started. Selecting type of the layer does not restrict geometry types which can be digitized in the vector map. In GRASS, it is possible to organize all sorts of geometry types (point, line and polygon) in one vector map. The type is only used to add the layer to the canvas, because QGIS requires a layer to have a specific type.

It is also possible to add layers to existing vector maps selecting one of the items described above from context menu of existing vector map.

In GRASS, it is possible to organize all sorts of geometry types (point, line and area) in one layer, because GRASS uses a topological vector model, so you don't need to select the geometry type when creating a new GRASS vector. This is different from shapefile creation with QGIS, because shapefiles use the Simple Feature vector model (see section *Naujų vektorinių sluoksnių kūrimas*).

22.12 Digitizing and editing a GRASS vector layer

GRASS vector layers can be digitized using the standard QGIS digitizing tools. There are however some particularities, which you should know about, due to

- GRASS topological model versus QGIS simple feature
- complexity of GRASS model
  - multiple layers in single maps
  - multiple geometry types in single map
  - geometry sharing by multiple features from multiple layers

The particularities are discussed in the following sections.

Save, discard changes, undo, redo

> **Įspėjimas**
>
> All the changes done during editing are immediately written to vector map and related attribute tables.

Changes are written after each operation, it is however, possible to do undo/redo or discard all changes when closing editing. If undo or discard changes is used, original state is rewritten in vector map and attribute tables.

There are two main reasons for this behaviour:

- It is the nature of GRASS vectors coming from conviction that user wants to do what he is doing and it is better to have data saved when the work is suddenly interrupted (for example, blackout)
- Necessity for effective editing of topological data is visualized information about topological correctness, such information can only be acquired from GRASS vector map if changes are written to the map.

Toolbar

The ‚Digitizing Toolbar' has some specific tools when a GRASS layer is edited:

22.1 Table : GRASS Digitizing Tools

Piktograma	Įrankis	Tikslas
	New Point	Digitize new point
	New Line	Digitize new line
	New Boundary	Digitize new boundary
	New Centroid	Digitize new centroid (label existing area)
	New Closed Boundary	Digitize new closed boundary

> **Patarimas**
>
> **Digitizing polygons in GRASS**
>
> If you want to create a polygon in GRASS, you first digitize the boundary of the polygon. Then you add a centroid (label point) into the closed boundary. The reason for this is that a topological vector model links the attribute information of a polygon always to the centroid and not to the boundary.

Category

Category, often called cat, is sort of ID. The name comes from times when GRASS vectors had only singly attribute „category“. Category is used as a link between geometry and attributes. A single geometry may have multiple categories and thus represent multiple features in different layers. Currently it is possible to assign only one category per layer using QGIS editing tools. New features have automatically assigned new unique category, except boundaries. Boundaries usually only form areas and do not represent linear features, it is however possible to define attributes for a boundary later, for example in different layer.

New categories are always created only in currently being edited layer.

It is not possible to assign more categories to geometry using QGIS editing, such data are properly represented as multiple features, and individual features, even from different layers, may be deleted.

Attributes

Attributes of currently edited layer can only be modified. If the vector map contains more layers, features of other layers will have all attributes set to ‚<not editable (layer #)>‘ to warn you that such attribute is not editable. The reason is, that other layers may have and usually have different set of fields while QGIS only supports one fixed set of fields per layer.

If a geometry primitive does not have a category assigned, a new unique category is automatically assigned and new record in attribute table is created when an attribute of that geometry is changed.

Patarimas

If you want to do bulk update of attributes in table, for example using ‚Field Calculator‘ (*Using the Field Calculator*), and there are features without category which you don't want to update (typically boundaries), you can filter them out by setting ‚Advanced Filter‘ to `cat is not null`.

Editing style

The topological symbology is essential for effective editing of topological data. When editing starts, a specialized ‚GRASS Edit‘ renderer is set on the layer automatically and original renderer is restored when editing is closed. The style may be customized in layer properties ‚Style‘ tab. The style can also be stored in project file or in separate file as any other style. If you customize the style, do not change its name, because it is used to reset the style when editing is started again.

Patarimas

Do not save project file when the layer is edited, the layer would be stored with ‚Edit Style‘ which has no meaning if layer is not edited.

The style is based on topological information which is temporarily added to attribute table as field ‚topo_symbol‘. The field is automatically removed when editing is closed.

Patarimas

Do not remove ‚topo_symbol‘ field from attribute table, that would make features invisible because the renderer is based on that column.

Pritraukimas

To form an area, vertices of connected boundaries must have **exactly** the same coordinates. This can be achieved using snapping tool only if canvas and vector map have the same CRS. Otherwise, due conversion from map coordinates to canvas and back, the coordinate may become slightly different due to representation error and CRS transformations.

Patarimas

Use layer's CRS also for canvas when editing.

Limitations

Simultaneous editing of multiple layers within the same vector at the same time is not supported. This is mainly due to the impossibility of handling multiple undo stacks for a single data source.

On Linux and macOS only one GRASS layer can be edited at time. This is due to a bug in GRASS which does not allow to close database drivers in random order. This is being solved with GRASS developers.

Patarimas

GRASS Edit Permissions

You must be the owner of the GRASS `MAPSET` you want to edit. It is impossible to edit data layers in a `MAPSET` that is not yours, even if you have write permission.

22.13 The GRASS region tool

The region definition (setting a spatial working window) in GRASS is important for working with raster layers. Vector analysis is by default not limited to any defined region definitions. But all newly created rasters will have the spatial extension and resolution of the currently defined GRASS region, regardless of their original extension and resolution. The current GRASS region is stored in the `$LOCATION/$MAPSET/WIND` file, and it defines north, south, east and west bounds, number of columns and rows, horizontal and vertical spatial resolution.

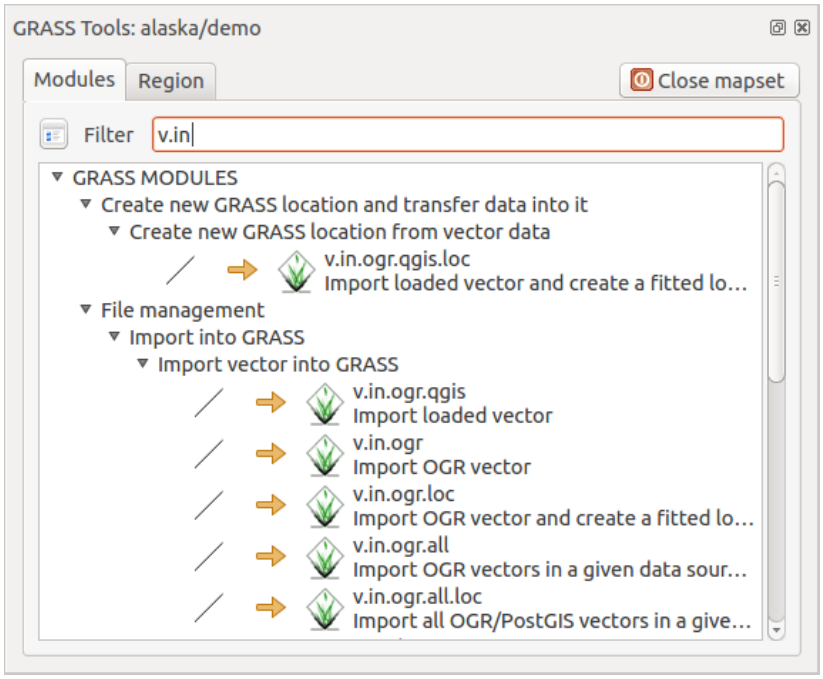
It is possible to switch on and off the visualization of the GRASS region in the QGIS canvas using the Display current GRASS region button.

The region can be modified in ‚Region' tab in ‚GRASS Tolls' dock widget. Type in the new region bounds and resolution, and click *Apply*. If you click on *Select the extent by dragging on canvas* you can select a new region interactively with your mouse on the QGIS canvas dragging a rectangle.

The GRASS module `g.region` provides a lot more parameters to define an appropriate region extent and resolution for your raster analysis. You can use these parameters with the GRASS Toolbox, described in section *The GRASS Toolbox*.

22.14 The GRASS Toolbox

The Open GRASS Tools box provides GRASS module functionalities to work with data inside a selected GRASS `LOCATION` and `MAPSET`. To use the GRASS Toolbox you need to open a `LOCATION` and `MAPSET` that you have write permission for (usually granted, if you created the `MAPSET`). This is necessary, because new raster or vector layers created during analysis need to be written to the currently selected `LOCATION` and `MAPSET`.



22.3 Fig. : GRASS Toolbox and Module Tree

22.14.1 Working with GRASS modules

The GRASS plugin provides a user-friendly toolbox with about 200 of the available GRASS modules and functionalities.

A complete list of GRASS modules available in the graphical Toolbox in QGIS version 3.44 is available in the GRASS wiki at https://grasswiki.osgeo.org/wiki/GRASS-QGIS_relevant_module_list.

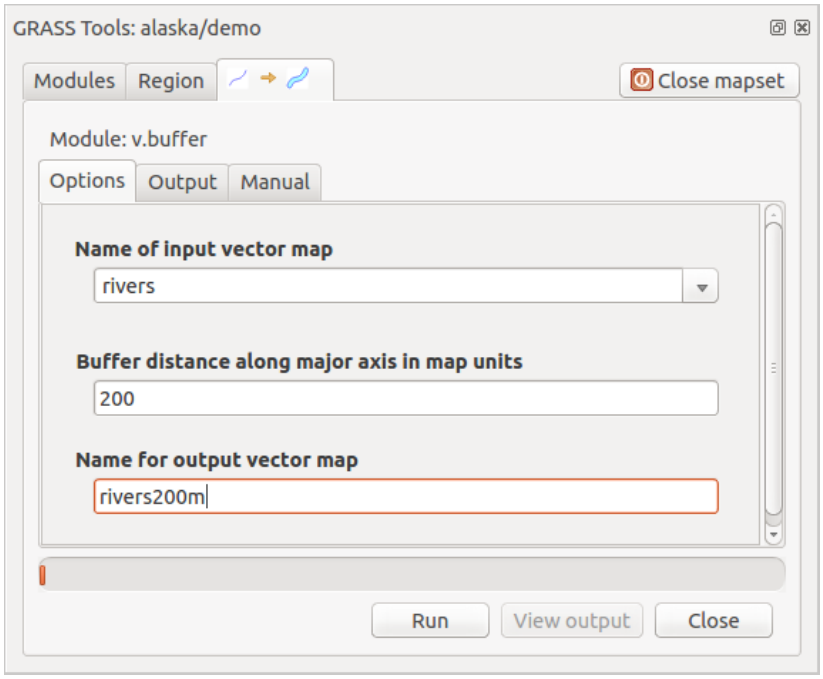
It is also possible to customize the GRASS Toolbox content. This procedure is described in section *Customizing the GRASS Toolbox*.

As shown in Fig. 22.3, you can look for the appropriate GRASS module using the thematically grouped *Modules Tree* or the searchable *Modules List* tab.

By clicking on a graphical module icon, a new tab will be added to the Toolbox dialog, providing three new sub-tabs: *Options*, *Output* and *Manual*.

Options

The *Options* tab provides a simplified module dialog where you can usually select a raster or vector layer visualized in the QGIS canvas and enter further module-specific parameters to run the module.

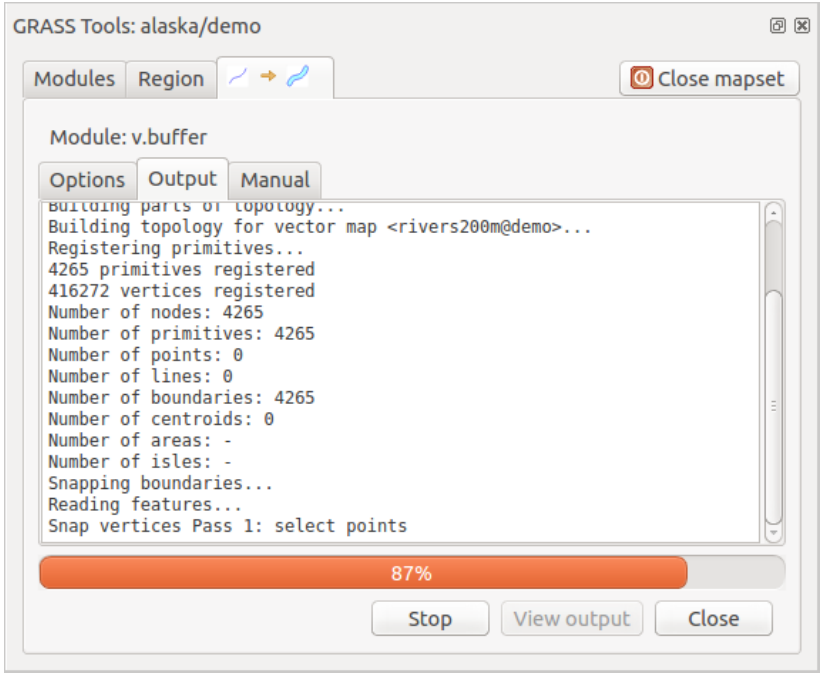


22.4 Fig. : GRASS Toolbox Module Options

The provided module parameters are often not complete to keep the dialog simple. If you want to use further module parameters and flags, you need to start the GRASS shell and run the module in the command line.

A new feature since QGIS 1.8 is the support for a *Show Advanced Options* button below the simplified module dialog in the *Options* tab. At the moment, it is only added to the module `v.in.ascii` as an example of use, but it will probably be part of more or all modules in the GRASS Toolbox in future versions of QGIS. This allows you to use the complete GRASS module options without the need to switch to the GRASS shell.

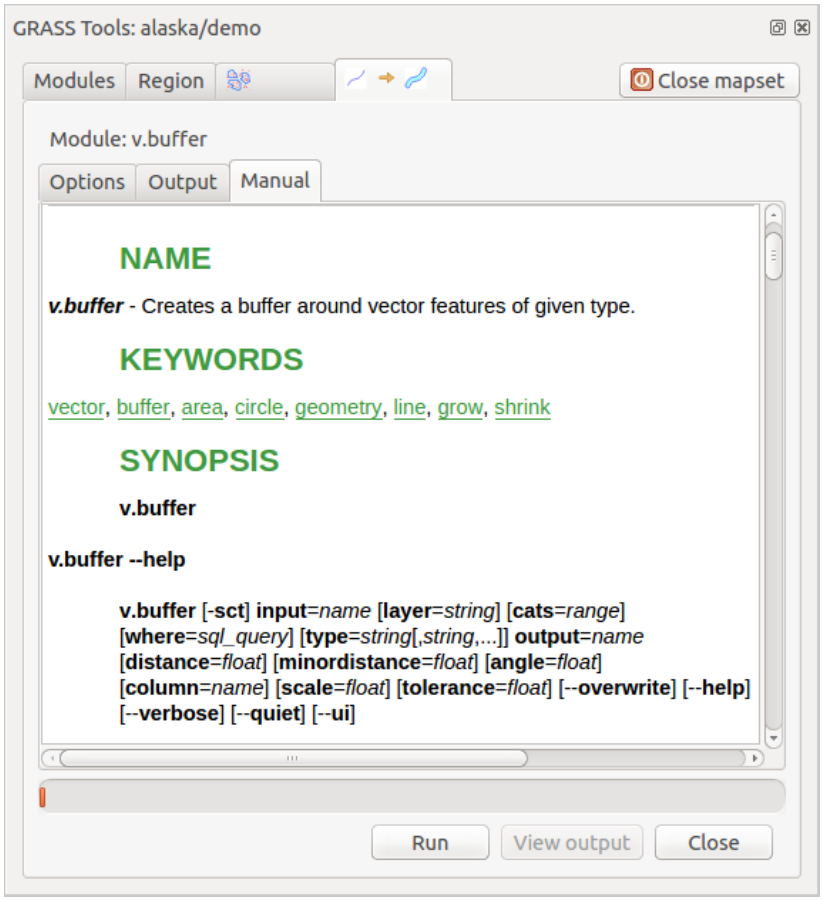
Išvestis



22.5 Fig. : GRASS Toolbox Module Output

The *Output* tab provides information about the output status of the module. When you click the *Run* button, the module switches to the *Output* tab and you see information about the analysis process. If all works well, you will finally see a `Successfully finished` message.

Manual



22.6 Fig. : GRASS Toolbox Module Manual

The *Manual* tab shows the HTML help page of the GRASS module. You can use it to check further module parameters and flags or to get a deeper knowledge about the purpose of the module. At the end of each module manual page, you see further links to the `Main Help index`, the `Thematic index` and the `Full index`. These links provide the same information as the module `g.manual`.

Patarimas

Display results immediately

If you want to display your calculation results immediately in your map canvas, you can use the ‚View Output' button at the bottom of the module tab.

22.14.2 GRASS module examples

The following examples will demonstrate the power of some of the GRASS modules.

Creating contour lines

The first example creates a vector contour map from an elevation raster (DEM). Here, it is assumed that you have the Alaska `LOCATION` set up as explained in section *Importing data into a GRASS LOCATION*.

- First, open the location by clicking the Open mapset button and choosing the Alaska location.
- Now open the Toolbox with the Open GRASS tools button.
- In the list of tool categories, double-click *Raster* ► *Surface Management* ► *Generate vector contour lines*.
- Now a single click on the tool **r.contour** will open the tool dialog as explained above (see *Working with GRASS modules*).
- In the *Name of input raster map* enter `gtopo30`.
- Type into the *Increment between Contour levels* the value 100. (This will create contour lines at intervals of 100 meters.)
- Type into the *Name for output vector map* the name `ctour_100`.
- Click *Run* to start the process. Wait for several moments until the message `Successfully finished` appears in the output window. Then click *View Output* and *Close*.

Since this is a large region, it will take a while to display. After it finishes rendering, you can open the layer properties window to change the line color so that the contours appear clearly over the elevation raster, as in *Vektorių savybių dialogas*.

Next, zoom in to a small, mountainous area in the center of Alaska. Zooming in close, you will notice that the contours have sharp corners. GRASS offers the **v.generalize** tool to slightly alter vector maps while keeping their overall shape. The tool uses several different algorithms with different purposes. Some of the algorithms (i.e., Douglas Peuker and Vertex Reduction) simplify the line by removing some of the vertices. The resulting vector will load faster. This process is useful when you have a highly detailed vector, but you are creating a very small-scale map, so the detail is unnecessary.

> **Patarimas**
>
> **The simplify tool**
>
> Note that QGIS has a *Vector* ► *Geometry Tools* ► *Simplify geometries* tool that works just like the GRASS **v.generalize** Douglas-Peuker algorithm.

However, the purpose of this example is different. The contour lines created by `r.contour` have sharp angles that should be smoothed. Among the **v.generalize** algorithms, there is Chaiken's, which does just that (also Hermite splines). Be aware that these algorithms can **add** additional vertices to the vector, causing it to load even more slowly.

- Open the GRASS Toolbox and double-click the categories *Vector* ► *Develop map* ► *Generalization*, then click on the **v.generalize** module to open its options window.
- Check that the ‚ctour_100' vector appears as the *Name of input vector*.
- From the list of algorithms, choose Chaiken's. Leave all other options at their default, and scroll down to the last row to enter in the field *Name for output vector map* ‚ctour_100_smooth', and click *Run*.
- The process takes several moments. Once `Successfully finished` appears in the output windows, click *View Output* and then *Close*.

- You may change the color of the vector to display it clearly on the raster background and to contrast with the original contour lines. You will notice that the new contour lines have smoother corners than the original while staying faithful to the original overall shape.

22.7 Fig. : GRASS module v.generalize to smooth a vector map

Patarimas

Other uses for r.contour

The procedure described above can be used in other equivalent situations. If you have a raster map of precipitation data, for example, then the same method will be used to create a vector map of isohyetal (constant rainfall) lines.

Creating a Hillshade 3-D effect

Several methods are used to display elevation layers and give a 3-D effect to maps. The use of contour lines, as shown above, is one popular method often chosen to produce topographic maps. Another way to display a 3-D effect is by hillshading. The hillshade effect is created from a DEM (elevation) raster by first calculating the slope and aspect of each cell, then simulating the sun's position in the sky and giving a reflectance value to each cell. Thus, you get sun-facing slopes lighted; the slopes facing away from the sun (in shadow) are darkened.

- Begin this example by loading the `gtopo30` elevation raster. Start the GRASS Toolbox, and under the Raster category, double-click to open *Spatial analysis* ► *Terrain analysis*.
- Then click **r.shaded.relief** to open the module.
- Change the *azimuth angle* 1,00 270 to 315.
- Enter `gtopo30_shade` for the new hillshade raster, and click *Run*.
- When the process completes, add the hillshade raster to the map. You should see it displayed in grayscale.
- To view both the hillshading and the colors of the `gtopo30` together, move the hillshade map below the `gtopo30` map in the table of contents, then open the *Properties* window of `gtopo30`, switch to the *Transparency* tab and set its transparency level to about 25%.

You should now have the `gtopo30` elevation with its colormap and transparency setting displayed **above** the grayscale hillshade map. In order to see the visual effects of the hillshading, turn off the `gtopo30_shade` map, then turn it back on.

Raster statistics in a vector map

The next example shows how a GRASS module can aggregate raster data and add columns of statistics for each polygon in a vector map.

- Again using the Alaska data, refer to *Importing data into a GRASS LOCATION* to import the `shapefiles/trees.shp` file into GRASS.
- Now an intermediate step is required: centroids must be added to the imported trees map to make it a complete GRASS area vector (including both boundaries and centroids).
- From the Toolbox, choose *Vector* ► *Manage features*, and open the module **v.centroids**.
- Enter as the *output vector map* ‚forest_areas' and run the module.
- Now load the `forest_areas` vector and display the types of forests - deciduous, evergreen, mixed - in different colors: In the layer *Properties* window, *Symbology* tab, choose from *Legend type* ‚Unique value' and set the *Classification field* to ‚VEGDESC'. (Refer to the explanation of the symbology tab in *Symbology Properties* of the vector section.)
- Next, reopen the GRASS Toolbox and open *Vector* ► *Vector update* by other maps.
- Click on the **v.rast.stats** module. Enter `gtopo30` and `forest_areas`.
- Only one additional parameter is needed: Enter *column prefix* `elev`, and click *Run*. This is a computationally heavy operation, which will run for a long time (probably up to two hours).
- Finally, open the `forest_areas` attribute table, and verify that several new columns have been added, including `elev_min`, `elev_max`, `elev_mean`, etc., for each forest polygon.

22.14.3 Customizing the GRASS Toolbox

Nearly all GRASS modules can be added to the GRASS Toolbox. An XML interface is provided to parse the pretty simple XML files that configure the modules' appearance and parameters inside the Toolbox.

A sample XML file for generating the module `v.buffer` (v.buffer.qgm) looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE qgisgrassmodule SYSTEM "http://mrcc.com/qgisgrassmodule.dtd">

<qgisgrassmodule label="Vector buffer" module="v.buffer">
        <option key="input" typeoption="type" layeroption="layer" />
        <option key="buffer"/>
        <option key="output" />
</qgisgrassmodule>
```

The parser reads this definition and creates a new tab inside the Toolbox when you select the module.

SKYRIUS 23

QGIS processing framework

23.1 Įvadas

This chapter introduces the QGIS processing framework, a geoprocessing environment that can be used to call native and third-party algorithms from QGIS, making your spatial analysis tasks more productive and easy to accomplish.

As a *Core plugin*, Processing is installed by default but you need to activate it:

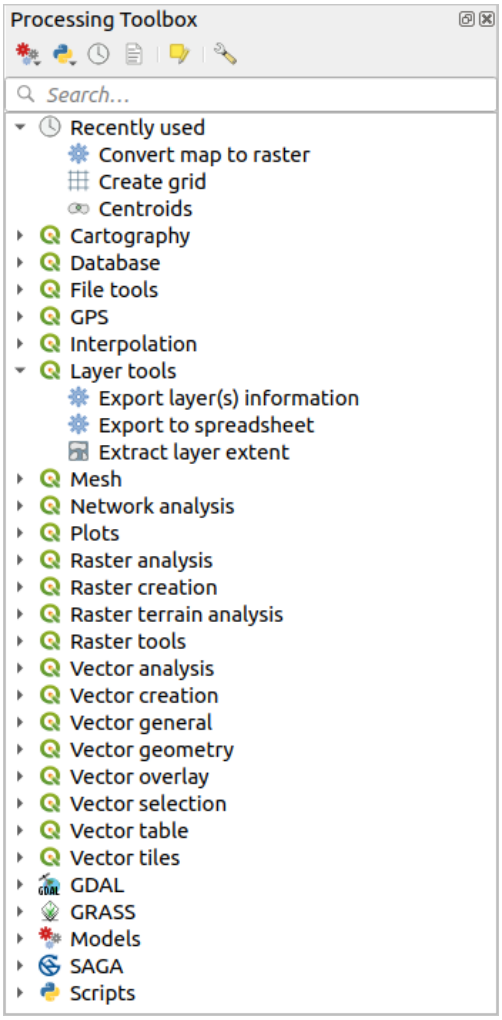
1. Go to *Plugins* ► *Manage and install plugins…*
2. Click on the *Installed* tab at the left
3. Check the box next to the *Processing* entry
4. Close the dialog.

   A *Processing* menu is now available in the top menu bar. From there you can reach the main components of this framework.

In the following sections, we will review how to use the graphical elements of this framework and make the most out of each one of them.

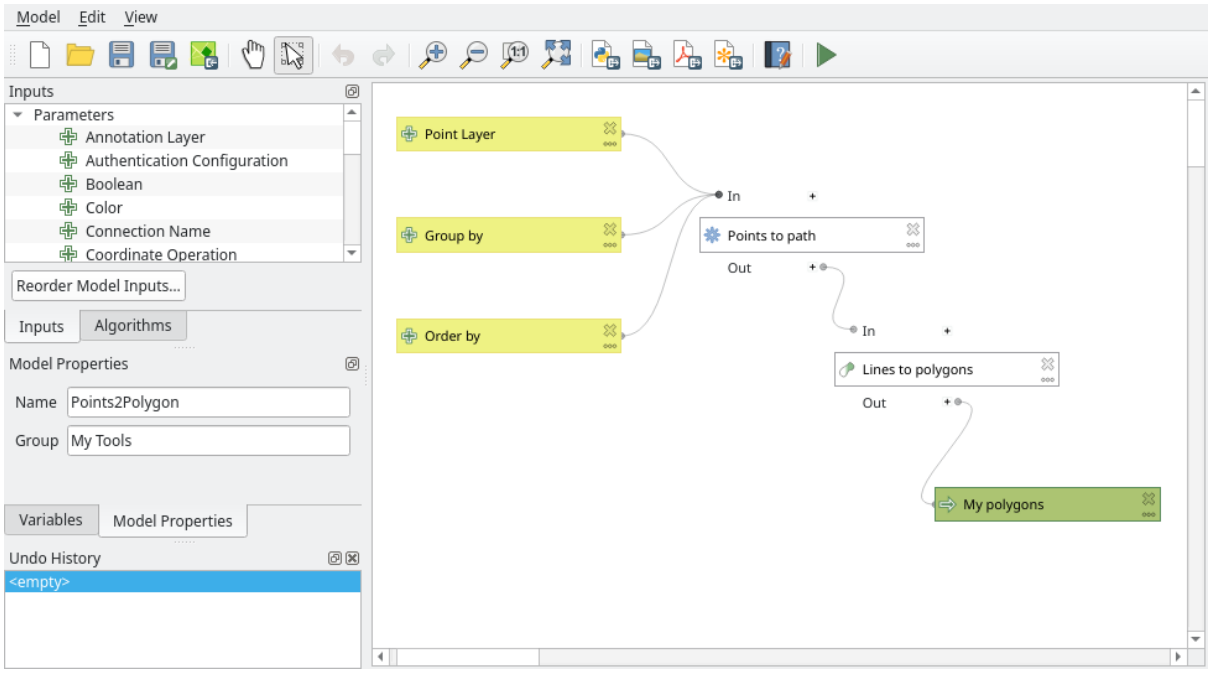
There are four basic elements in the framework GUI, which are used to run algorithms for different purposes. Choosing one tool or another will depend on the kind of analysis that is to be performed and the particular characteristics of each user and project. All of them (except for the batch processing interface, which is called from the toolbox or the algorithm execution dialog, as we will see) can be accessed from the *Processing* menu item (you will see more entries; the remaining ones are not used to execute algorithms and will be explained later in this chapter).

- The *Toolbox*: The main element of the GUI, it is used to execute a single algorithm or run a batch process based on that algorithm.



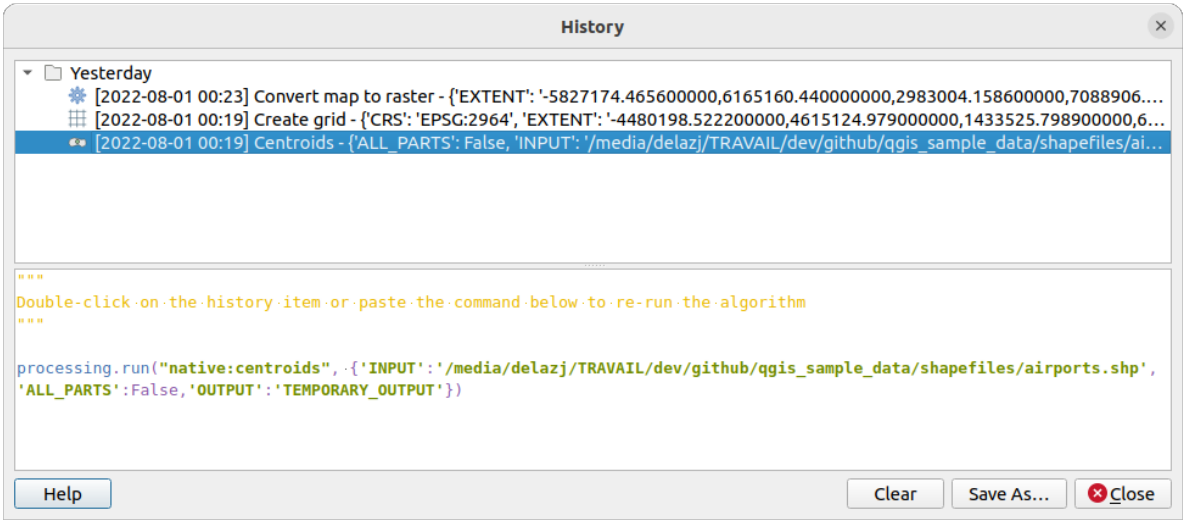
23.1 Fig. : Apdorojimo įrankinės

- The *Model Designer*: Several algorithms can be combined graphically using the modeler to define a workflow, creating a single process that involves several subprocesses.



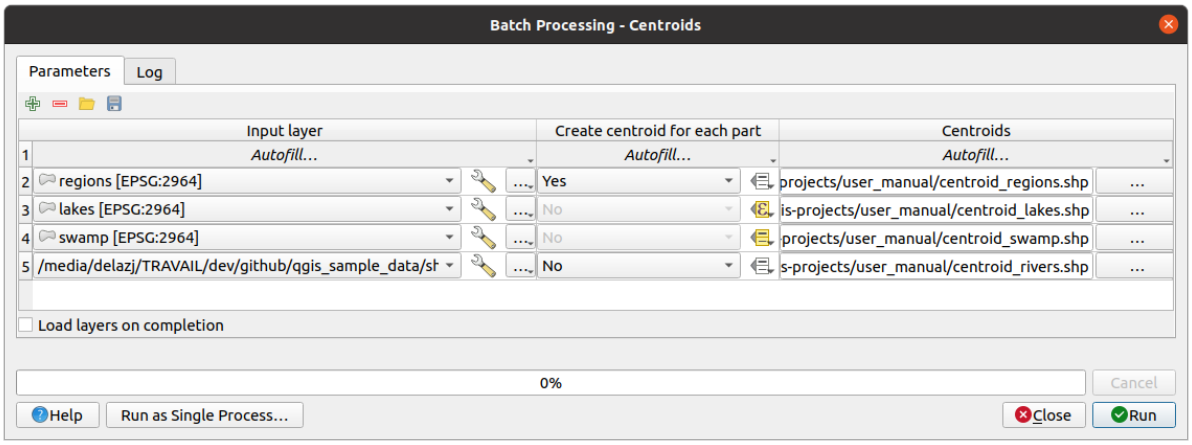
23.2 Fig. : Apdorojimo modeliuotojas

- The *History* manager: All actions performed using any of the aforementioned elements are stored in a history file and can be later easily reproduced using the history manager.



23.3 Fig. : Processing History

- The *Batch Processing* interface: This interface allows you to execute batch processes and automate the execution of a single algorithm on multiple datasets.



23.4 Fig. : Batch Processing interface

In the following sections, we will review each one of these elements in detail.

23.2 Apdorojimo karkaso konfigūravimas

Apdorojimo parinkčių meniu (*Nustatymai*► *Parinktys* ► *Apdorojimas* kortelė) leidžia jums konfigūruoti, kaip veikia algoritmai. Konfigūracijos parametrų struktūroje yra atskiri blokai, kuriuos jūs galite parinkti kairėje dialogo pusėje.

23.2.1 Bendra

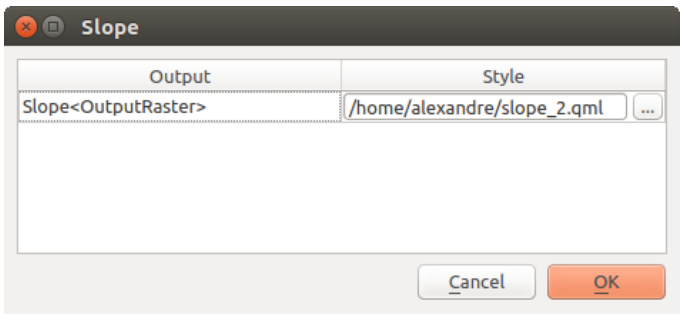
Skiltyje *Bendra* yra numatytieji nustatymai, valdantys algoritmų dialogų bei įvesties ar išvesties parametrų elgseną. Kai kuriuos nustatymus visgi galima permušti algoritmų lygyje konkrečiam *algoritmo paleidimui* ar *individualiems parametrams*.

- *Numatytas išvesties rastro sluoksnio išplėtimas* pagal nutylėjimą yra `tif`
- *Numatytasis išvesties vektoriaus sluoksnio išplėtimas* pagal nutylėjimą yra `gpkg`
- *Neteisingų geoobjektų filtravimas* vykdant algoritmą:
  - *Nefiltruoti (geresnė greitaveika)*: apdorojami visi geoobjektai (su tinkamomis ir netinkamomis geometrijomis), bet rezultatas gali būti klaidingas priklausomai nuo to, kaip geometrijos netinkamumas įtakoja veiksmus
  - *Praleisti (ignoruoti) geoobjektus su netinkamomis geometrijomis*, tai reiškia, kad bus apdorotas tik jūsų duomenų aibės poaibis (geoobjektai su tinkama geometrija)
  - *Stabdyti algoritmo vykdymą, kai geometrija yra netinkama*: jums reikės sekti ir taisyti netinkamas geometrijas, jei norėsite, kad algoritmas apdorotų visą sluoksnį. Algoritmai, tokie kaip *Tikrinti tvarkingumą* ar *Fix geometries* jums gali padėti tai padaryti.

  Netinkamų geoobjektų filtravimo nustatymą galima permušti algoritmo vykdymo metu kiekvienam įvesties rinkiniui.
- *Palikti dialogą atvertą po algoritmo įvykdymo*. Kai algoritmo vykdymas baigiamas ir jo išvesties sluoksniai įkeliami į QGIS projektą, algoritmo dialogas uždaromas. Įjunkite šią parinktį, jei norite, kad jis liktų atvertas (kad dar kartą paleistumėte algoritmą su kitokiais parametrais ar įdėmiau peržvelgtumėte išvestį žurnalo kortelėje).
- *Maksimalus gijų skaičius*

- *Output folder* for non temporary outputs: If no folder path is provided for the Processing execution outputs, this is the folder in which they will be saved. Default is `processing` under the user home directory (e.g., `C:\Users\<YOURUSERNAME>\processing\` on Windows).
- *Override temporary output folder path*: Temporary outputs are saved by default in the `tmp` folder on the machine. This option helps you set a different place for storage.
- *Pre-execution script* and *Post-execution script*. These parameters point to files that contain scripts written using the processing scripting functionality, explained in the section covering scripting and the console.
- *Prefer output filename for layer names*. The name of each resulting layer created by an algorithm is defined by the algorithm itself. In some cases, a fixed name might be used, meaning that the same output name will be used, no matter which input layer is used. In other cases, the name might depend on the name of the input layer or some of the parameters used to run the algorithm. If this checkbox is checked, the name will be taken from the output filename instead. Notice that, if the output is saved to a temporary file, the filename of this temporary file is usually a long and meaningless one intended to avoid collision with other already existing filenames.
- *Results group name*. If you want to obtain all processing result layers in a group in the *Layers* panel, set a group name for this parameter. The group may exist already or not. QGIS will add all output layers to such a group. By default, this parameter is empty, so all output layers are added to different places in the *Layers* panel, depending on the item that is active when running an algorithm. Note that output layers will be loaded to the *Layers* panel only if *Open output file after running algorithm* is checked in the algorithm dialog.
- *Show algorithms with known issues*: By default, QGIS avoids display of broken algorithms (generally from third-party providers). If checked, they will be available in the Processing toolbox, with a warning icon and a tooltip explaining they have issues. Use at your own risks.
- *Show feature count for output vector layers*. Since calculating the feature count might take some time in certain data formats, this option is off by default.
- *Show layer CRS definition in selection boxes*
- *Show tooltip when there are disabled providers*
- *Style for line layers*, *Style for point layers*, *Style for polygons layers* and *Style for raster layers* are used for setting the default rendering style for output layers (that is, layers generated by processing algorithms). Just create the style you want using QGIS, save it to a file, and then enter the path to that file in the settings so the algorithms can use it. Whenever a layer is loaded by Processing and added to the QGIS canvas, it will be rendered with that style.

  Rendering styles can be configured individually for each algorithm and each one of its outputs. Just right-click on the name of the algorithm in the toolbox and select *Edit rendering styles for outputs*. You will see a dialog like the one shown next.



23.5 Fig. : Rendering Styles

  Select the style file (`.qml`) that you want for each output and press *OK*.
- *Warn before executing if parameter CRS's do not match*: By default, QGIS native algorithms (i.e. the ones listed under the *Menus* ► *QGIS (native C++)* group) transparently reproject the input layers to the first one's CRS

before execution. Check this option to get a notification from the other tools that do not suppport reprojection, when the inputs CRS are not identical. Third-party providers are not concerned.

23.2.2 Menus

The *Menus* block controls whether an algorithm, script or model (built-in or provided by plugins) should be made available through a dedicated menu or toolbar (along with the Processing Toolbox). For each item of each provider, you can:

- *Add button in toolbar*, making it available in the *Processing Algorithms* toolbar
- assign an *Icon* to the algorithm
- set a *Menu path*: the algorithm will then be available through an existing or a custom menu, e.g. `Vect&or/MyTopAlgorithms` will add a shortcut in a *MyTopAlgorithms* sub-menu of the default *Vector* menu.

Restart QGIS to apply the settings. At any time, your changes can be *Reset to defaults*.

23.2.3 Models and Scripts

In the *Models* and *Scripts* blocks, a default path, set under the active *User profile* folder, is provided for storing models and scripts respectively. As for the other options, you can modify the path, e.g., to lead to a remote or shared folder for all your users.

23.2.4 Providers

You will also find a block for algorithm *Providers*. This is the place where installed providers expose their settings.

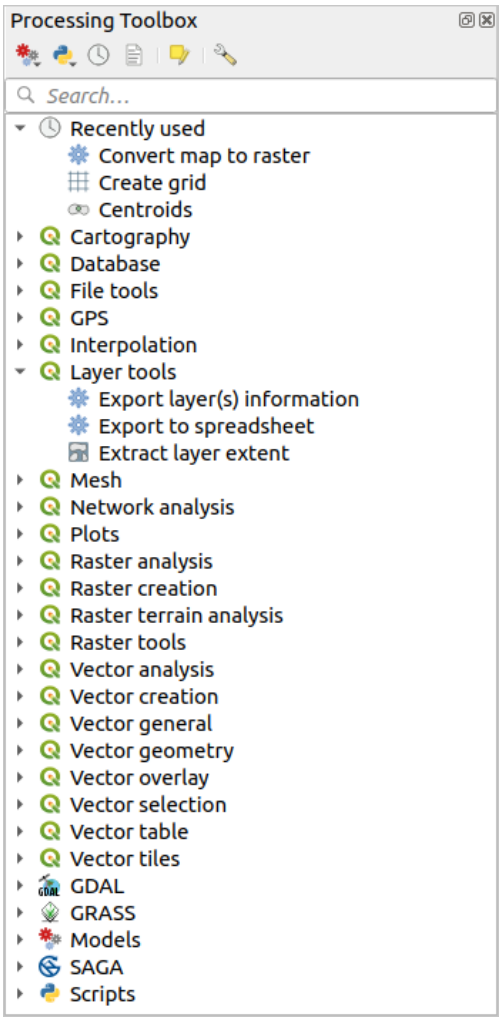
By default, QGIS is installed with two algorithm providers:

- *GDAL* whose algorithms you may *Activate* (or not) in this dialog. Read more about the *GDAL algorithms*.
- *GRASS* whose algorithms require the installation of the *GRASS GIS Processing Provider* from the *Plugins Manager*. You could then set some dedicated settings such as:
  - *For raster layers, use r.external (faster) instead of r.in.gdal*
  - *For vector layers, use v.external (faster) instead of v.in.ogr*
  - *Location of GRASS docs*: by default, and depending on your OS, QGIS utilizes a hardcoded list of paths for finding the local GRASS documentation and opening it whenever you hit the *Help* button of a GRASS algorithm dialog. Filling this option will override the default path, allowing you to use a custom location, or remote documentation.
  - *Log console output*
  - *Log execution commands*

Installed plugins that provide Processing algorithms may also have their provider listed in this group, with custom settings.

23.3 Įrankinė

Apdorojimo įrankinė yra pagrindinis apdorojimo GUI elementas, kurį jūs labiausiai tikėtina, kad naudosite kasdieniniame darbe. Jis rodo sąrašą prieinamų **algoritmų**, sugrupuotų į skirtingus blokus, vadinamų *Tiekėjai*, bei savus **modelius** bei **scenarijus**, kuriuos jūs galite pridėti ir išplėsti esamus įrankius. Taigi įrankinė yra taškas, per kurį jūs juos pasiekiate ir paleidžiate, nesvarbu ar kaip atskirus procesus ar kaip paketinius darbus, paleidžiančius tą patį algoritmą kelis kartus su skirtingomis įvestimis.



23.6 Fig. : Apdorojimo įrankinės

Tiekėjus galima (de)aktyvuoti *Apdorojimo nustatymų dialoge*. Pagal nutylėjimą aktyvūs tik tiekėjai, kuriems nereikia trečiųjų šalių aplikacijų (t.y. tie, kurių veikimui reikia tik QGIS elementų). Algoritmams, kuriems reikia išorinių aplikacijų, gali prireikti papildomo konfigūravimo. Tiekėjų konfigūravimas aprašomas šio vadovo *kitame skyriuje*.

Viršutinėje įrankinės dialogo dalyje rasite įrankius, kurie leidžia:

- dirbti su Modeliais: *Kurti naują modelį…*, *Atidaryti esamą modelį…* ir *Pridėti modelį į įrankinę…*;
- dirbti su Scenarijais: *Kurti naują scenarijų…*, *Kurti naują scenarijų pagal šabloną…*, *Atidaryti esamą scenarijų…* ir *Pridėti scenarijų į įrankinę…*;
- atverti Istorijos skydelį;
- atverti Rezultatų vaizdo skydelį;

- perjungti įrankinę į *keitimo vietoje režimą* naudojant mygtuką Keisti geoobjektus vietoje: rodomi tik tik algoritmai, kurie tinka būtų vykdomais aktyviame sluoksnyje nesukurdami naujo sluoksnio;
- atverti dialogą Parinktys.

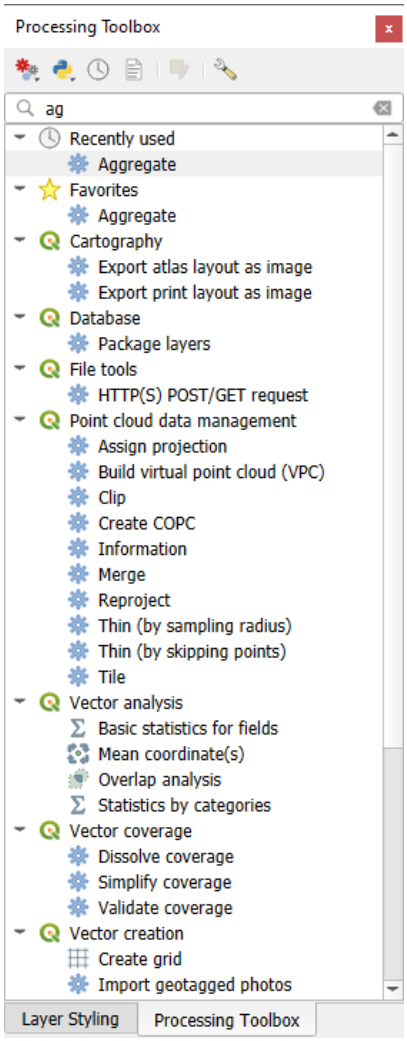
Po šia įrankine yra laukelis *Paieška…*, kuris leidžia jums lengvai rasti reikiamus įrankius. Į šį laukelį galite įvesti bet kokį žodį ar frazę. Pastebėtina, kad jums rašant, rodomų algoritmų, modelių ir scenarijų skaičius mažėja, rodant tik tuos, kurių pavadinimuose ar raktažodžiuose yra jūsų įrašytas tekstas.

Algoritmų sąrašo viršuje rodomi Neseniai naudoti ir Parankiniai įrankiai, taip padarant jų pakartotinį vykdymą patogesniu. Sąrašas Neseniai naudoti rodomas pagal nutylėjimą, jame rasite paskutinius vykdytus algoritmus. Sąrašas Parankiniai sukuriamas tada, kai jūs rankiniu būdu pridedate į jį algoritmą.

Galite spausti dešinį pelės mygtuką ant bet kurio algoritmo įrankinėje ir kontekstiniame meniu pasirinkti:

- *Vykdyti…*, kad paleistumėte algoritmą.
- *Vykdyti kaip paketinį procesą…*, kad vykdytumėte algoritmą paketiniame režime.
- *Keisti išvesties braižymo stilius…*, kad keistumėte algoritmo išvesčių braižymo stilius.
- *Pridėti į parankinius*, kad pridėtumėte algoritmą į Parankinių sąrašą.
- *Išimti iš parankinių*, kad išimtumėte algoritmą iš Parankinių sąrašo.

Priešingai nei Neseniai naudotų sąrašas, Parankinių sąrašas lieka nekintantis ir jo neįtakoja tai, kokie algoritmai buvo vykdomi.

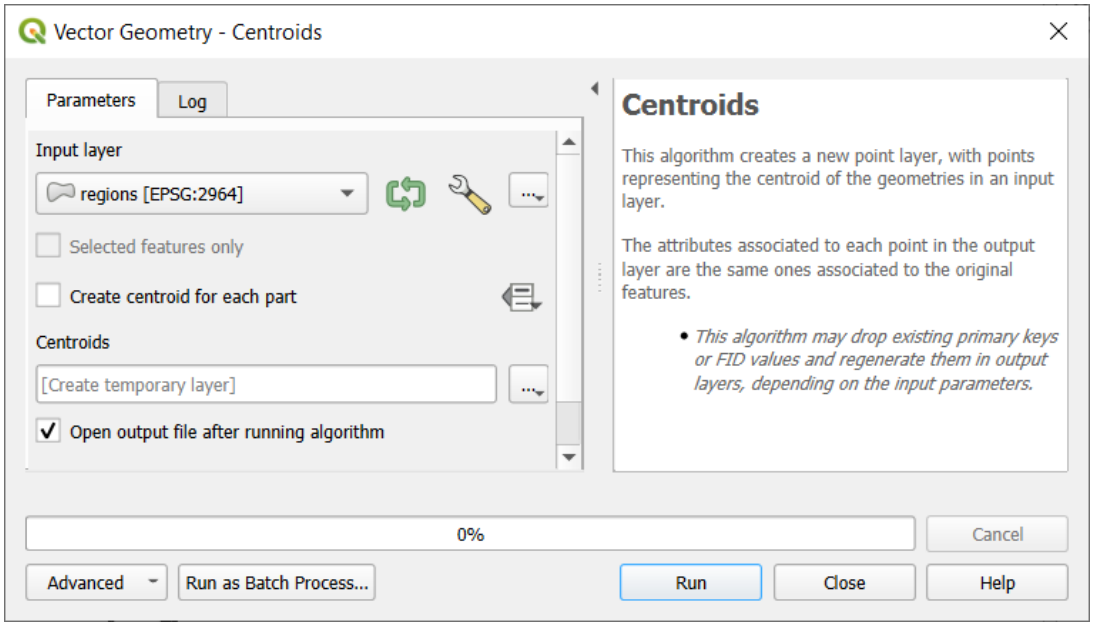


23.7 Fig. : Apdorojimo įrankinė, rodanti paieškos rezultatus

Kad įvykdytumėte įrankį, tiesiog du kartus spauskite jo pavadinimą įrankinėje.

23.3.1 The algorithm dialog

Once you double-click on the name of the algorithm that you want to execute, a dialog similar to that in the Fig. 23.8 below is shown (in this case, the dialog corresponds to the `Centroids` algorithm). Dialog title will include the group name from which the algorithm originates.



23.8 Fig. : Algorithm Dialog - Parameters

The dialog shows two tabs (*Parameters* and *Log*) on the left part, the algorithm description on the right, and a set of buttons at the bottom.

Parameter types

The *Parameters* tab is used to set the input values that the algorithm needs to be executed. It shows a list of input values and configuration parameters to be set. It of course has a different content, depending on the requirements of the algorithm to be executed, and is created automatically based on those requirements.

Patarimas

Setting your own default values for algorithm parameters

Algorithm dialogs open with some parameters prefilled with values from QGIS installation. It is however possible to set *your own default values* for specific algorithm parameters so that they are used at algorithm startup.

Although the number and type of parameters depend on the characteristics of the algorithm, the structure is similar for all of them. The parameters found in the table can be of one of the following types.

- A **vector layer**, to select from a list of all vector layers available (currently opened) in QGIS. You can also use unloaded layers: press the … button on the widget right-hand side, and select:
  - *Select file…*: selects file on disk using the Operating System file explorer
  - *Browse for layer…*: opens the *Browser panel*, allowing to take the layers directly from database sources (PostgreSQL, SQL Server, Oracle, …), web services (WFS, AFS, …) or files on disk.

23.9 Fig. : Vector input widget

Patarimas

Drag-and drop as convenient alternative for layer selection

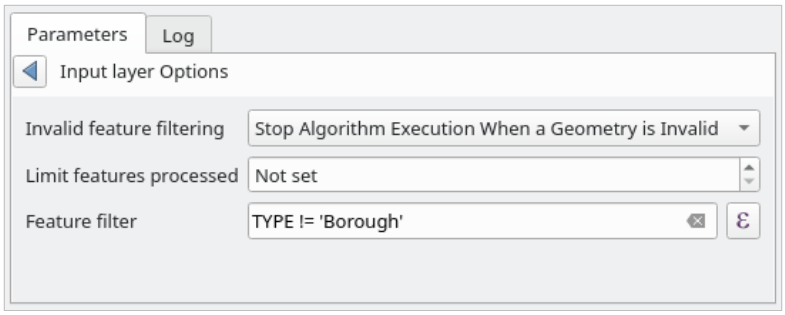
Besides selecting from the drop-down menu or browsing sources, you can also drag and drop a layer from the *Layers* or *Browser* panel, or from the Operating System file explorer directly into the combobox.

Pastaba

By default, the layer widget shows the CRS of the layer along with its name. If you do not want to see this additional information, you can disable this functionality in the Processing Settings dialog, unchecking the *General* ► *Show layer CRS definition in selection boxes* option.

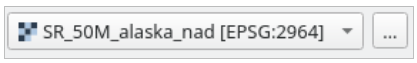
The vector input widget also has following features:

- an iterator button: If toggled, the algorithm will be executed iteratively on each one of its features, instead of just once for the whole layer, producing as many outputs as times the algorithm is executed. This allows for automating the process when all features in a layer have to be processed separately. If the algorithm contains several input vectors you can iterate over, the iteration will be processed only on the first toggled parameter, in the order parameters are declared in the algorithm.
- Advanced options button to adjust settings to use for that specific parameter. These settings concern:
  - *Invalid feature filtering*: allows the *default method* for handling features with invalid geometries to be overridden
  - *Limit features processed*: optional limit on number of features processed from the source
  - *Feature filter*: allows to enter an expression to subset the layer dynamically when running the tool, avoiding the need for separate steps to set layer filters or create layer subsets.



23.10 Fig. : Advanced options for vector input widget

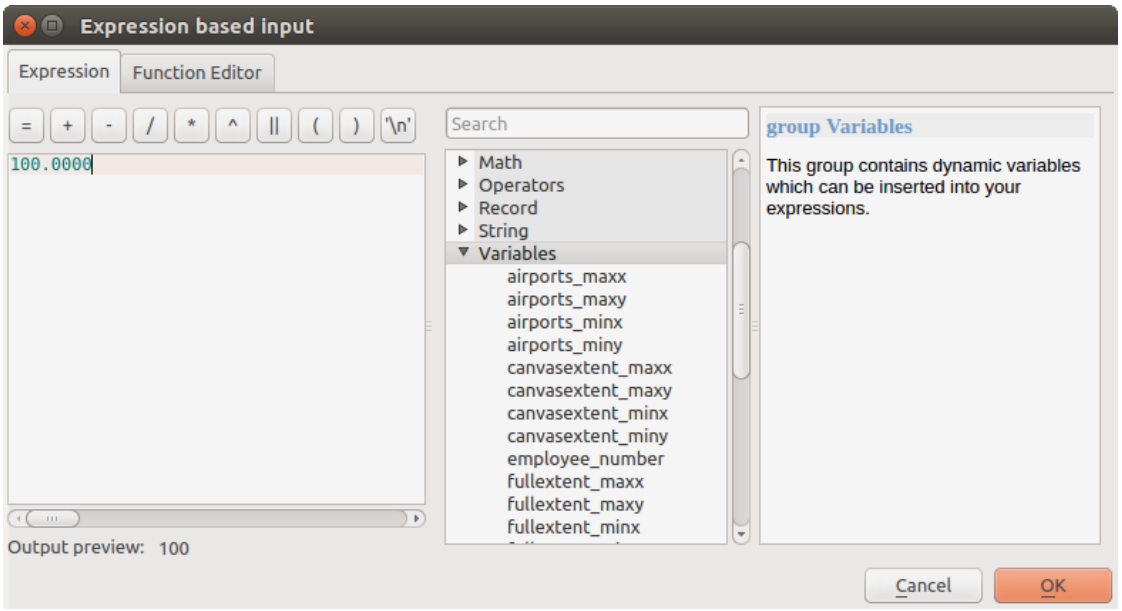
  - It is also possible to limit the algorithm execution on the vector layer to its *Selected features only*.
- A **table**, to select from a list of all available in QGIS. Non-spatial tables are loaded into QGIS like vector layers, and use the *same widget*.
- A **raster layer**, to select from a list of all raster layers available in QGIS. The selector contains as well a … button on its right-hand side, to let you select filenames that represent layers currently not loaded in QGIS.



23.11 Fig. : Raster input widget

- An **option**, to choose from a selection list of possible options.

- A **numerical value**, to be introduced in a spin box. In some contexts (when the parameter applies at the feature level and not at the layer's), you will find a Data-defined override button by its side, allowing you to open the *expression builder* and enter a mathematical expression to generate variable values for the parameter. Some useful variables related to data loaded into QGIS can be added to your expression, so you can select a value derived from any of these variables, such as the cell size of a layer or the northernmost coordinate of another one.



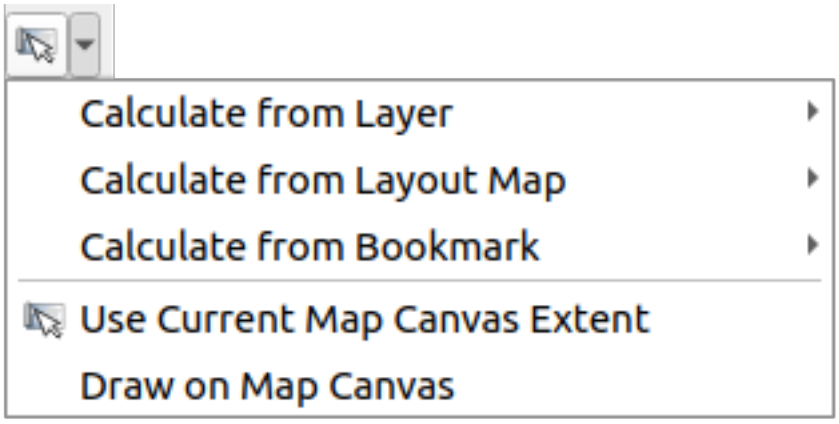
23.12 Fig. : Išraiška paremta įvestis

- A **range**, with min and max values to be introduced in two text boxes.
- A **text string**, to be introduced in a text box.
- A **field**, to choose from the attributes table of a vector layer or a single table selected in another parameter.
- A **coordinate reference system**. You can select it among the recently used ones from the drop-down list or from the *CRS selection* dialog that appears when you click on the button on the right-hand side.

> **Patarimas**
>
> **Drag-and-drop as convenient shortcut for CRS selection**
>
> You can also drag a layer from the Layers panel and drop it onto the CRS widget. The widget will automatically set its CRS to match the dropped layer's.

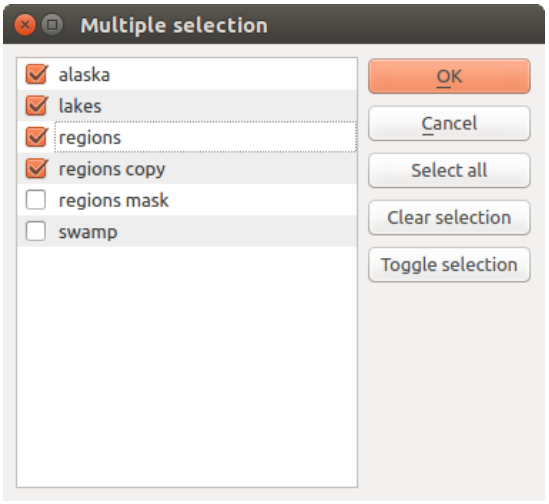
- An **extent**, a text box defining a rectangle through its corners coordinate in the format `xmin, xmax, ymin, ymax`. Press the Set to current map canvas extent button to use the map canvas extent. Clicking the arrow on the right-hand side of the value selector, a pop-up menu will appear, giving you options to:
  - *Calculate from layer* ►: fills the text box with the coordinates of the bounding box of a layer to select among the loaded ones
  - *Calculate from layout map* ►: fills the text box with the coordinates of a map item selected from a layout in the current project
  - *Calculate from bookmark* ►: fills the text box with the coordinates of a saved bookmark
  - *Use current map canvas extent*

– *Draw on canvas*: the parameters window will hide itself, so you can click and drag onto the canvas. Once you have defined the extent rectangle, the dialog will reappear, containing the values in the extent text box.



23.13 Fig. : Extent selector

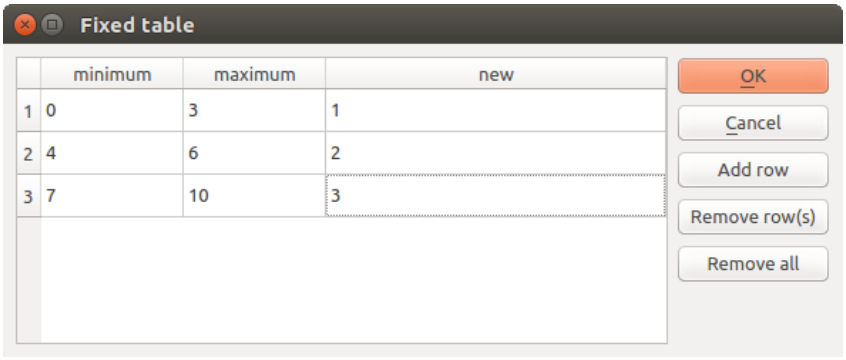
- A **list of elements** (whether raster or vector layers, tables, fields) to select from. Click on the … button at the left of the option to see a dialog like the following one. Multiple selection is allowed and when the dialog is closed, number of selected items is displayed in the parameter text box widget.



23.14 Fig. : Multiple Selection

- A **small table** to be edited by the user. These are used to define parameters like lookup tables or convolution kernels, among others.

  Click on the button on the right side to see the table and edit its values.



23.15 Fig. : Fixed Table

Depending on the algorithm, the number of rows can be modified or not by using the buttons on the right side of the window.

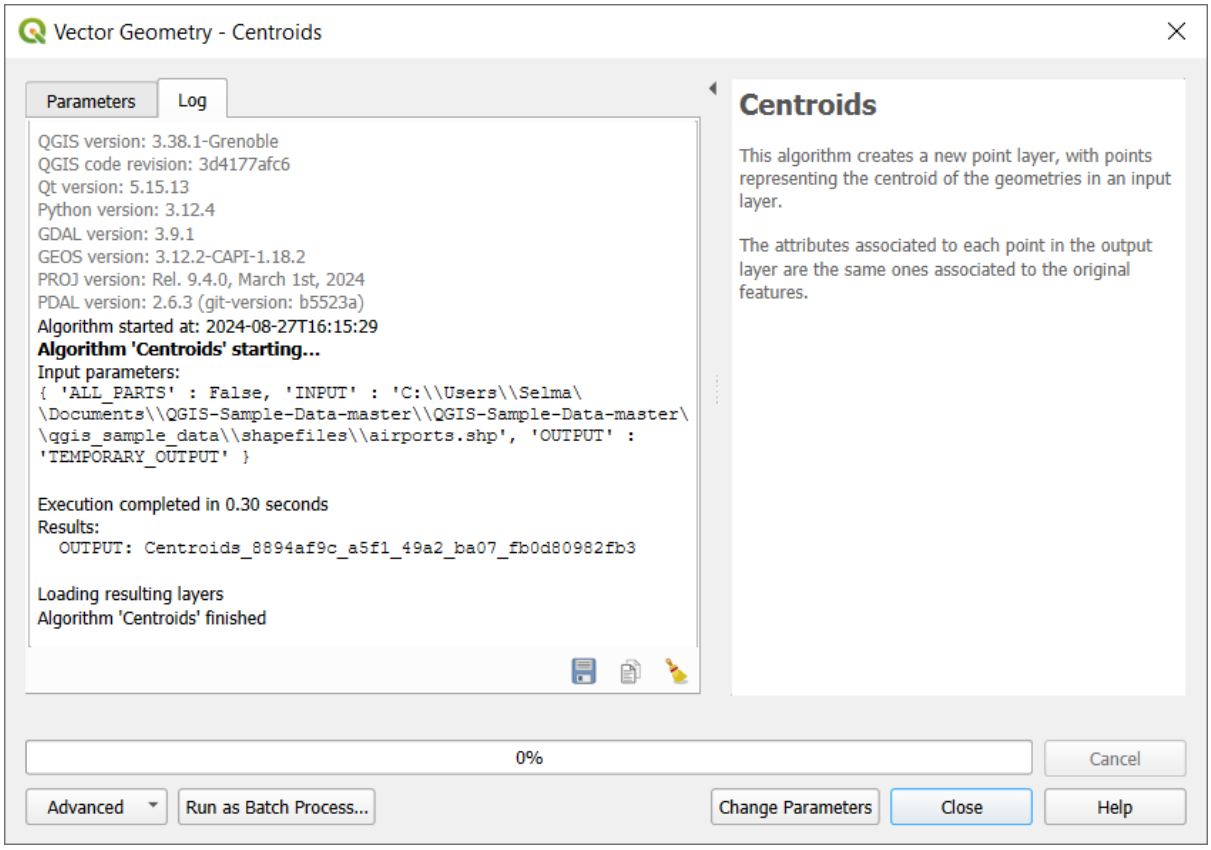
Pastaba

Some algorithms require many parameters to run, e.g. in the *Raster calculator* you have to specify manually the cell size, the extent and the CRS. You can avoid to choose all the parameters manually when the algorithm has the `Reference layers` parameter. With this parameter you can choose the reference layer and all its properties (cell size, extent, CRS) will be used.

Logging the execution

Along with the *Parameters* tab, there is another tab named *Log* (see Fig. 23.16 below). Information provided by the algorithm during its execution is written in this tab, allowing you to track the execution as well as being aware and having more details about the algorithm as it runs. You can directly click on the names of output files, folders, or HTML files listed in the log. Doing so will open the folder containing the generated file and automatically select it. Information on algorithm execution is also output in the *View* ► *Panels* ► *Log Messages Panel*.

Notice that not all algorithms write information to the *Log* tab, and many of them might run silently without producing any output other than the final files. Check the *Log Messages Panel* in that case.



23.16 Fig. : Algorithm Dialog - Log

At the bottom of the *Log* tab you will find buttons to *Save Log to File*, *Copy Log to Clipboard* and *Clear Log*. These are particularly handy when you have checked the *Keep dialog open after running algorithm* in the *General* part of the Processing options.

Other tools

On the right hand side of the dialog you will find a short description of the algorithm, which will help you understand its purpose and its basic ideas. If such a description is not available, the description panel will not be shown.

For a more detailed help file, which might include description of every parameter it uses, or examples, you will find a *Help* button at the bottom of the dialog bringing you to the *Processing algorithms documentation* or to the provider documentation (for some third-party providers).

The *Advanced* ► menu provides functions to reuse the configuration defined in the dialog without running the algorithm:

- *Algorithm Settings…*: allows to override processing settings for the current algorithm execution. More details at *Override algorithm settings*.
- *Copy as Python Command*: allows for easy copying of the equivalent *PyQGIS command* to run the tool using the parameters defined in the dialog
- *Copy as qgis_process Command*: allows for easy generation of *qgis_process command*, including its environment settings like the distance units, area units, ellipsoid, and any tricky parameter values like GeoPackage outputs with specific layers
- *Copy as JSON*: all the settings of the command are copied in a `JSON` format, ready to be consumed by qgis_process. This is a convenient way to see the expected format of the commands, even for complex

parameters (like TIN interpolation parameters). You can store these easily and then restore them later by pasting the values.

- *Paste Settings* in a `JSON` format

The *Run as Batch Process…* button triggers the *batch processing mode* allowing to configure and run multiple instances of the algorithm with a variety of parameters. A *Run as Single Process…* button helps you switch back from the batch mode.

When an algorithm execution finishes (either successfully or not), a new button *Change Parameters* is shown as long as the *Log* tab is active.

Override algorithm settings

Triggered from within the *Advanced* drop-down menu at the bottom of an algorithm dialog, the *Algorithm Settings…* shows a panel allowing users to control general processing settings which apply to that algorithm execution only. It is intended to be a place where a user can override their *global processing settings* on an ad-hoc basis without having to change their usual default settings.

Settings that can be overridden are:

- *Invalid feature filtering*: unlike the existing per-parameter setting override for this, setting the handling method here will apply to **ALL inputs** for the algorithm
- *Calculation settings*, such as *Distance units* and *Area units* to use for distance/area measurements
- *Environment settings*, such as *Temporary folder* and *Number of threads to use*

A note on projections

Processing algorithm execution are always performed in the input layer coordinate reference system (CRS). Due to QGIS's on-the-fly reprojecting capabilities, although two layers might seem to overlap and match, that might not be true if their original coordinates are used without reprojecting them onto a common coordinate system. Whenever you use more than one layer as input to a *QGIS native algorithm*, whether vector or raster, the layers will all be reprojected to match the coordinate reference system of the first input layer.

This is however less true for most of the external applications whose algorithms are exposed through the processing framework as they assume that all of the layers are already in a common coordinate system and ready to be analyzed.

By default, the parameters dialog will show a description of the CRS of each layer along with its name, making it easy to select layers that share the same CRS to be used as input layers. If you do not want to see this additional information, you can disable this functionality in the Processing settings dialog, unchecking the *Show layer CRS definition in selection boxes* option.

If you try to execute an algorithm using as input two or more layers with unmatching CRSs, a warning dialog will be shown. This occurs thanks to the *Warn before executing if layer CRS's do not match* option.

You still can execute the algorithm, but be aware that in most cases that will produce wrong results, such as empty layers due to input layers not overlapping.

Patarimas

Use Processing algorithms to do intermediate reprojection

When an algorithm can not successfully perform on multiple input layers due to unmatching CRSs, use QGIS internal algorithm such as *Reproject layer* to perform layers' reprojection to the same CRS before executing the algorithm using these outputs.

23.3.2 Data objects generated by algorithms

Data objects generated by an algorithm can be of different types. They can be a layer (vector with or without geometry, raster, mesh, point cloud, ...), stored as a plain file on disk or in a database, text-based files such as `qml` style file or graphics `.html` file, a folder, ...

The parameters dialog will contain a widget corresponding to the needed output, where you can type the output channel to use for saving it. An output channel contains the information needed to save the resulting object somewhere. In the most usual case, you will save it to a file, but in the case of vector layers, and when they are generated by native algorithms (algorithms not using external applications) you can also save to a PostgreSQL, GeoPackage or SpatiaLite database, or a memory layer.

The **output** parameter for setting the result of the algorithm execution provides a text box for filling the path to the output result, and a drop-down ... button for additional options (the availability depends on the algorithm you run):

- *Skip Output*: if you are not interested in a given output of the algorithm
- *Create Temporary Layer* (`TEMPORARY_OUTPUT`): the output is stored in a vector *temporary scratch layer*.
- *Save to File...*: you will be prompted with a save file dialog, where you can select the desired file path. Supported file extensions are shown in the file format selector of the dialog, depending on the kind of output and the algorithm.

  The format of the output is defined by the filename extension. The supported formats depend on what is supported by the algorithm itself. To select a format, just select the corresponding file extension (or add it, if you are directly typing the file path instead). If the extension of the file path you entered does not match any of the supported formats, a default extension will be appended to the file path, and the file format corresponding to that extension will be used to save the layer or table. Default extensions are `.dbf` for tables, `.tif` for raster layers and `.gpkg` for vector layers. These can be modified in the setting dialog, selecting any other of the formats supported by QGIS.

  You can set a default folder for output data objects. Go to the settings dialog (you can open it from the *Settings* ► *Options* ► *Processing* menu), and in the *General* group, you will find a parameter named *Output folder*. This output folder is used as the default path in case you type just a filename with no path (i.e., `myfile.shp`) when executing an algorithm.
- *Save to a Temporary File*: if you do not enter any filename in the output text box (or select the corresponding option in the context menu), the result will be saved as a temporary file in the corresponding default file format, and it will be deleted once you exit QGIS (take care with that, in case you save your project and it contains temporary layers). The file is saved in the default `TEMP` folder on the machine, if *not overridden*.
- *Save to Geopackage...*: the output layer will be saved as a named table of a new or existing GeoPackage database
- *Save to Database Table...*: the output layer will be saved as a new table in a database such as GeoPackage, SpatiaLite, PostgreSQL or MS SQL Server. A connection should already exist. You'll need to provide the name of the new table and depending on the type of target database, select a schema,
- *Append to Layer...*: adds features output by the algorithm to an existing layer. Upon selecting the target layer, a field mapping panel opens, allowing to manually set how fields from the output layer are mapped to the target layer's fields.

- *Save to Directory*: Use a new or existing directory for storing the algorithm outputs
- *Save to a Temporary Directory*: creates a directory within the set `TEMP` folder for storing the algorithm outputs. The files will be deleted once you exit QGIS.

When running an algorithm that uses a vector layer in iterative mode, the entered file path is used as the base path for all generated files, which are named using the base name and appending a number representing the index of the iteration. The file extension (and format) is used for all such generated files.

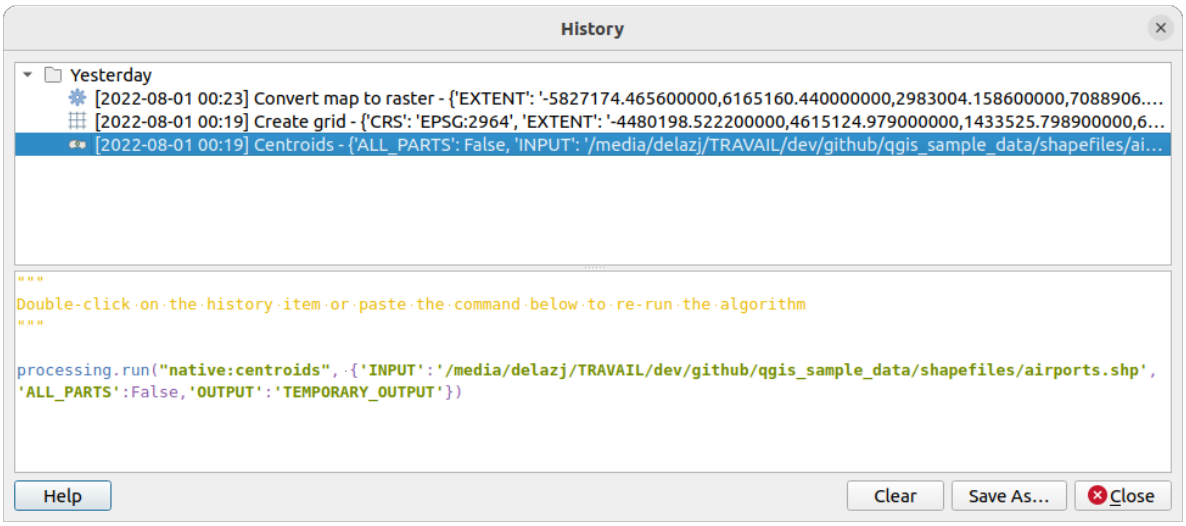
Algorithms also generate graphics and text as HTML files. These results are shown at the end of the algorithm execution in the *Results Viewer* dialog. This dialog will keep the results produced by any algorithm during the current session, and can be shown at any time by selecting *Processing* ► *Results Viewer* from the QGIS main menu.

For layer-based outputs, it is possible to *Open output file after running algorithm* directly in QGIS, as a new entry in the *Layers* panel. By default, the files are opened.

23.4 The history manager

23.4.1 The processing history

Every time you execute an algorithm, information about the process is stored in the history manager. The date and time of the execution are saved, along with the parameters used, making it is easy to track and control all the work that has been developed using the Processing framework, and to reproduce it.



23.17 Fig. : Istorija

Process information is kept as a command-line expression, even if the algorithm was launched from the toolbox. This makes it useful for those learning how to use the command-line interface, since they can call an algorithm using the toolbox and then check the history manager to see how it could be called from the command line.

Right-click on a row and you should be able to:

- *Copy as Python Command*: allows for easy copying of the equivalent *PyQGIS command* run from the dialog. Same as the code displayed below the commands list.
- *Copy as qgis_process Command*: allows for easy generation of *qgis_process command*, including its environment settings like the distance units, area units, ellipsoid, and any tricky parameter values like GeoPackage outputs with specific layers
- *Copy as JSON*: all the settings of the command are copied in a `JSON` format, ready to be consumed by qgis_process. This is a convenient way to see the expected format of the commands, even for complex parameters (like TIN interpolation parameters). You can store these easily and then restore them later by pasting the values over an algorithm dialog.
- *Create Test...* using the concerned algorithm and parameters, following instructions at Processing README file.

Apart from browsing the entries in the registry, you can also re-execute processes by simply double-clicking on the entry. The algorithm dialog then opens with parameters already set, and you can change any of them to fit your needs and re-run the algorithm.

The *History* dialog also provides a convenient way to contribute to the consolidation of the testing infrastructure of QGIS Processing algorithms and scripts.

23.4.2 The processing log

The history dialog only contains the execution calls, but not the information produced by the algorithm when executed. That information is written to the QGIS log (*View* ► *Panels* ► *Log Messages Panel*).

Third-party algorithms are usually executed by using their command-line interfaces, which communicate with the user via the console. Although that console is not shown, usually a full dump of it is written to the log each time you run one of those algorithms. To avoid cluttering the log with that information, you can disable it for each provider in the settings dialog.

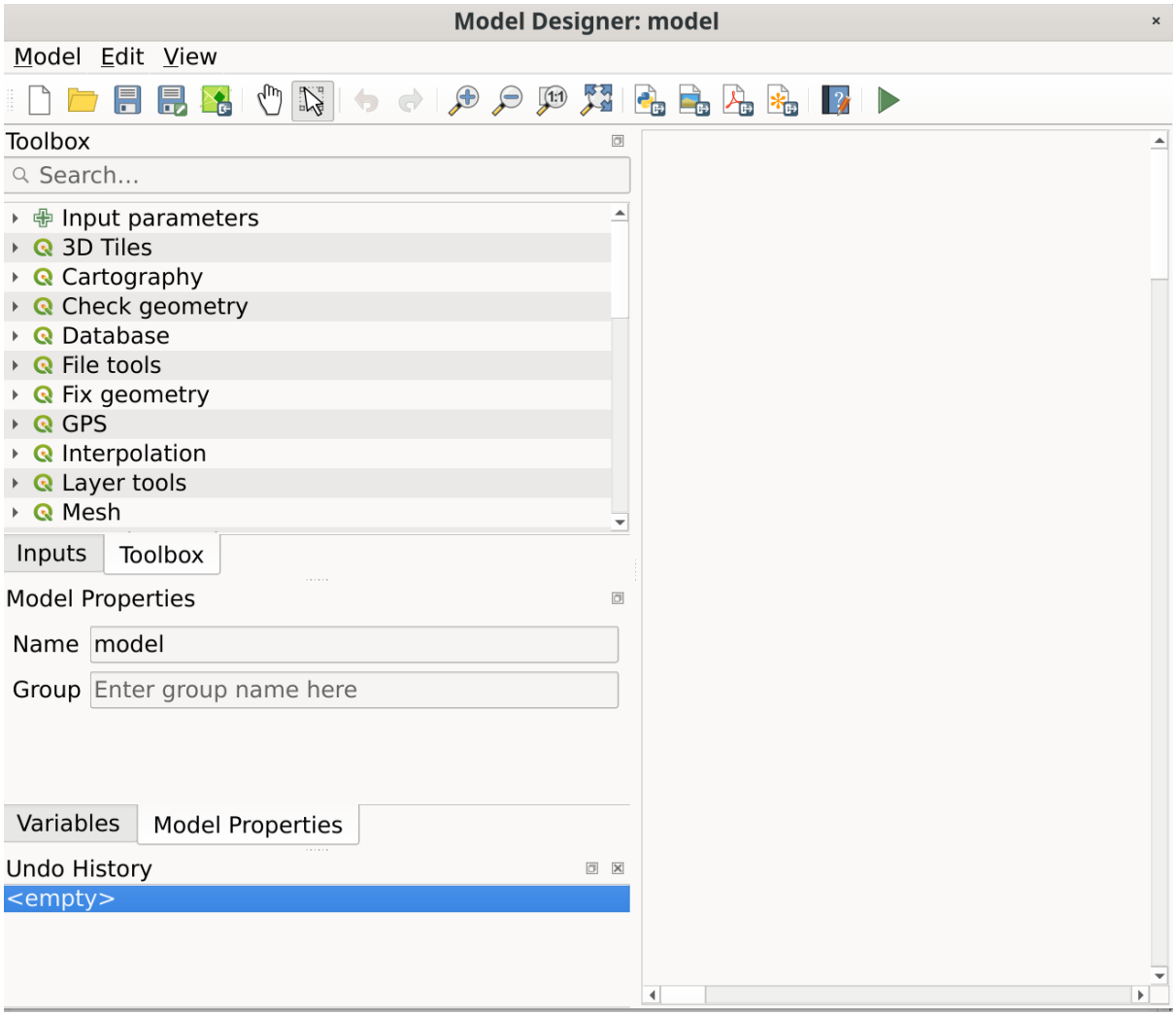
Some algorithms, even if they can produce a result with the given input data, output comments or additional information to log when they detect potential problems with the data, in order to warn you. Make sure you check those messages in the log if you get unexpected results.

23.5 Modelių dizaineris

Modelių dizaineris leidžia jums kurti sudėtingus modelius naudojant paprastą ir lengvai naudojamą sąsają. Dirbant su GIS, dauguma analizės veiksmų nėra izoliuoti, o formuoja veiksmų grandinę. Naudojant modelio dizainerį, tokia veiksmų grandinė gali būti supakuota į vieną procesą, taip padarant jį paprastesniu vykdymui vėliau su kitomis įvestimis. Nepriklausomai nuo to, kiek jame yra skirtingų žingsnių ir algoritmų, modelis vykdomas kaip vienas algoritmas, taip taupant laiką ir pastangas.

Modelių dizainerį galima atidaryti per apdorojimo meniu (*Apdorojimas* ► *Modelių dizaineris*).

23.5.1 Modelių dizainerio sąsaja



23.18 Fig. : Modelio dizaineris

Pagrindinėje modeliuotoje dalyje yra darbo drobė, kurioje konstruojama modelio struktūra ir darbo eiga.

Dialogo viršuje yra skirtingi meniu, o *Navigacijos* įrankinė teikia įvairius įrankius.

Modelio meniu

Užrašas	Kombinacij	Navigacijos įrankinė	Aprašymas
Tikrinti modelį			Tikrina ar algoritmai ir modelyje naudojamos įvestys egzistuoja. Patogu prieš išleidžiant modelį.
Vykdyti modelį…	`F5`	☑	Vykdo modelį
Vykdyti parinktus žingsnius…	`Shift+F5`		Vykdo tik parinktus modelio žingsnius, leidžia vykdyti modelio poaibį
Perrikiuoti modelio įvestis…			Nustato tvarką, kuria algoritmo dialoge naudotojui pateikiamos įvesty.
Perrikiuoti išvesties sluoksnius…			Nustato konkrečią tvarką, kurią modelio išvestys turi naudoti įkeliant rezultatus į projektą.
Atverti modelį…	`Ctrl+O`	☑	Atveria `.model3` failą keitimui ar vykdymui
Įrašyti modelį	`Ctrl+S`	☑	Įrašo modelį į diską kaip `.model3` failą
Įrašyti modelį kaip…	`Ctrl+Shi`	☑	Įrašo modelį į diską kaip naują `.model3` failą
Įrašyti modelį projekte		☑	Įtraukia modelio failą į projekto failą, taip padarant jį prieinamu dalinantis projekto failu.
Keisti modelio pagalbą…		☑	Naudotojo sąsaja, skirta dokumentuoti modelį, algoritmus, parametrus, išvestis bei autorių ir versijas
Eksportuoti ►			
► *Eksportuoti kaip piešinį…*		☑	Įrašo modelio grafinį dizainą kaip piešinio failą (iliustravimo tikslais)
► *Eksportuoti kaip PDF…*			Įrašo modelio grafinį dizainą į `PDF` failą (iliustravimo tikslais)
► *Eksportuoti kaip SVG…*			Įrašo modelio grafinį dizainą į `SVG` failą (iliustravimo tikslais)
► *Eksportuoti kaip scenarijaus algoritmą…*		☑	Sukuria python skripto failą su modelio instrukcijomis

Keitimo meniu

Užrašas	Kombinacij	Navigacijos įrankinė	Aprašymas
Pažymėti viską	`Ctrl+A`		Pažymi dizaineryje visus modelio komponentus
Pritraukti pažymėtus komponentus prie tinklelio			pritraukia ir sulygiuoja elementus pagal tinklelį
Pakartoti	`Ctrl+Y`	☑	Atstatyti paskutinį atšauktą veiksmą. Taipogi žiūrėkite skydelį *Atstatyti/pakartoti*.
Atšaukti	`Ctrl+Z`	☑	Atšaukti paskutinį pakeitimą. Taipogi žiūrėkite skydelį *Atstatyti/pakartoti*.
Iškirpti	`Ctrl+X`		Iškerpa iš modelio komponentų pažymėjimą.
Kopijuoti	`Ctrl+C`		Kopijuoja modelio komponentų pažymėjimą.
Įkelti	`Ctrl+V`		Įkelia iškirptą arba nukopijuotą modelio komponentų rinkinį į kitą arba tą patį modelį. Pažymėti komponentai išlaiko savo pradines savybes ir komentarus.
Trinti pažymėtus komponentus	`Del`		Išima komponentą iš modelio.
Pridėti grupę			Pridėta grupę į foną susijusių komponentų, kad vizualiai juos sugrupuotų. Ypatingai naudinga dideliuose modeliuose siekiant išlaikyti aiškią eigą.

Vaizdo meniu

Užrašas	Kombinacij	Navigacijos įrankinė	Aprašymas
Didinti iki ►			Priartina parinktos grupės apimtį
Didinti	`Ctrl++`	☑	
Mažinti	`Ctrl+-`	☑	
Didinti iki 100%	`Ctrl+1`	☑	
Pilnai padidinti	`Ctrl+0`	☑	Rodo visus dabartinės dizainerio drobės komponentus
Rodyti komentarus			Rodo modelio dizainerio visų algoritmų ar įvesčių komentarus
Įjungti pritraukimą			
Perjungti skydelio matomumą	`Ctrl+Tab`		Įjungia arba išjungia dizainerio *skydelius*

Skydeliai

The left part of the window is a section with five panels that can be used to add new elements to the model:

1. *Inputs*: all the *input parameters* that could shape your model
2. *Toolbox*: here you can find available *Processing algorithms*. It also includes input parameters. Use *Search…* box to find algorithms or inputs.
3. *Model Properties*: specify the name (required) of the model and the group in which it will be displayed in the *Processing Toolbox*
4. *Variables*: Models can contain dedicated *variables* that are unique and only available to them. These variables can be accessed by any expression used within the model. They are useful to control algorithms within a model and control multiple aspects of the model by changing a single variable. The variables can be viewed and modified in the *Variables* panel.
5. *Undo History*: this panel will register everything that happens in the modeler, making it easy to cancel things you did wrong.

About available algorithms

Some algorithms that can be executed from the *Processing* ► *Toolbox* do not appear in the list of available algorithms when you are designing a model. To be included in a model, an algorithm must have the correct semantic. If an algorithm does not have such a well-defined semantic (for instance, if the number of output layers cannot be known in advance), then it is not possible to use it within a model, and it will not appear in the list of algorithms that you can find in the modeler dialog. On the other hand some algorithms are specific to the modeler. Those algorithms are located within the group ‚Modeler Tools'.

23.5.2 Creating a model

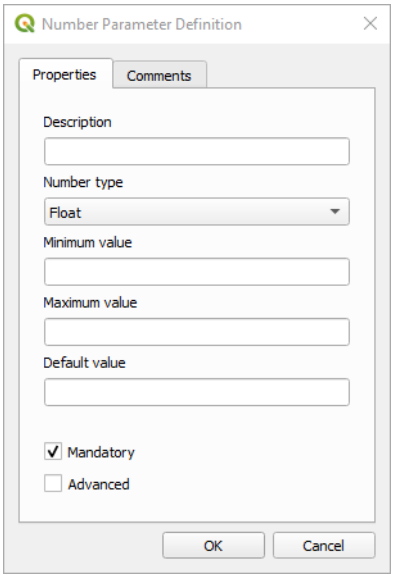
Creating a model involves two basic steps:

1. *Definition of necessary inputs*. These inputs will be added to the parameters window, so the user can set their values when executing the model. The model itself is an algorithm, so the parameters window is generated automatically as for all algorithms available in the Processing framework.
2. *Definition of the workflow*. Using the input data of the model, the workflow is defined by adding algorithms and selecting how they use the defined inputs or the outputs generated by other algorithms in the model.

Definition of inputs

The first step is to define the inputs for the model. They are found in the *Inputs* panel on the left side of the modeler window. Hovering with the mouse over the inputs will show a tooltip with additional information. For a full list of available parameters in modeler and their correspondance for scripting, please read *Apdorojimo algoritmų įvesties ir išvesties parametrai*.

When double-clicking on an element, a dialog is shown that lets you define its characteristics. Depending on the parameter, the dialog will contain at least one element (the description, which is what the user will see when executing the model). For example, when adding a numerical value, as can be seen in the next figure, in addition to the description of the parameter, you have to set a default value and the range of valid values.



23.19 Fig. : Model Parameters Definition

You can define your input as mandatory for your model by checking the Mandatory option and by checking the Advanced checkbox you can set the input to be within the Advanced section. This is particularly useful when the model has many parameters and some of them are not trivial, but you still want to choose them.

For each added input, a new element is added to the modeler canvas.

23.20 Fig. : Model Parameters

You can also add inputs by dragging the input type from the list and dropping it at the position where you want it in the modeler canvas. If you want to change a parameter of an existing input, just double click on it, and the same dialog will pop up.

When using a model within another model, the inputs and outputs necessary will be displayed in the canvas.

Definition of the workflow

In the following example we will add two inputs and two algorithms. The aim of the model is to copy the elevation values from a DEM raster layer to a line layer using the Drape algorithm, and then calculate the total ascent of the line layer using the Climb Along Line algorithm.

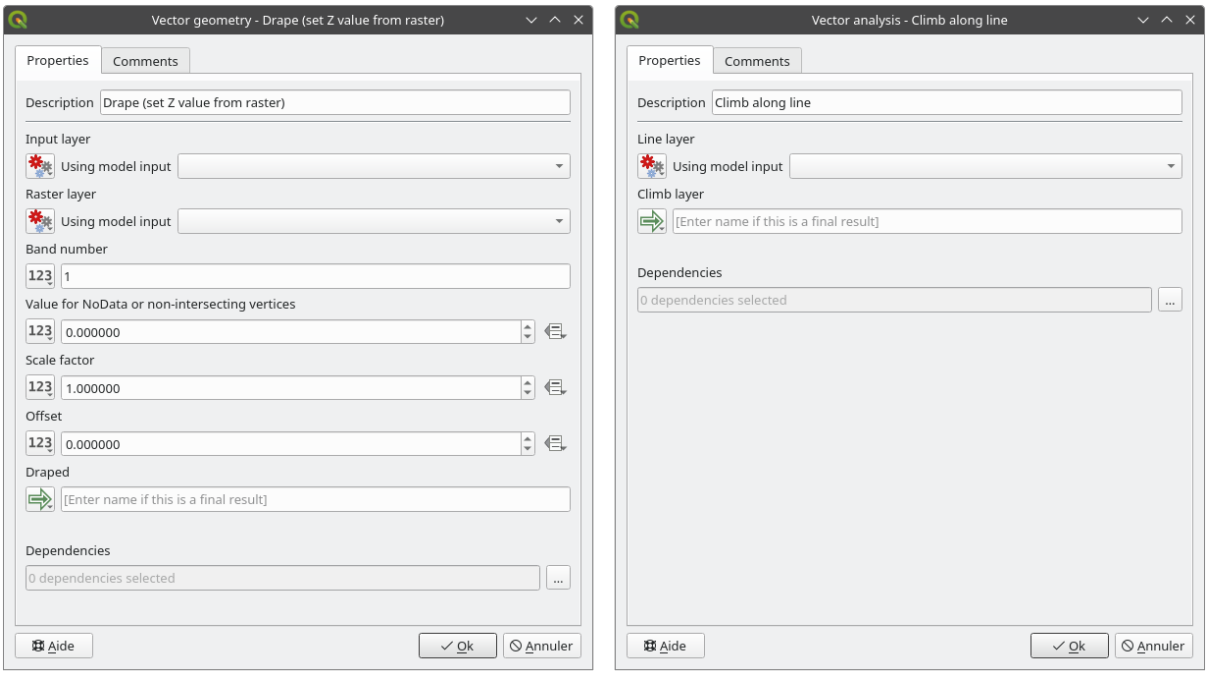
In the *Inputs* tab, choose the two inputs as Vector Layer for the line and Raster Layer for the DEM. We are now ready to add the algorithms to the workflow.

Algorithms can be found in the *Toolbox* panel, grouped much in the same way as they are in the Processing toolbox.

23.21 Fig. : Model Inputs

To add an algorithm to a model, double-click on its name or drag and drop it, just like for inputs. As for the inputs you can change the description of the algorithm and add a comment. When adding an algorithm, an execution dialog

will appear, with a content similar to the one found in the execution panel that is shown when executing the algorithm from the toolbox. The following picture shows both the `Drape (set Z value from raster)` and the `Climb along line` algorithm dialogs.

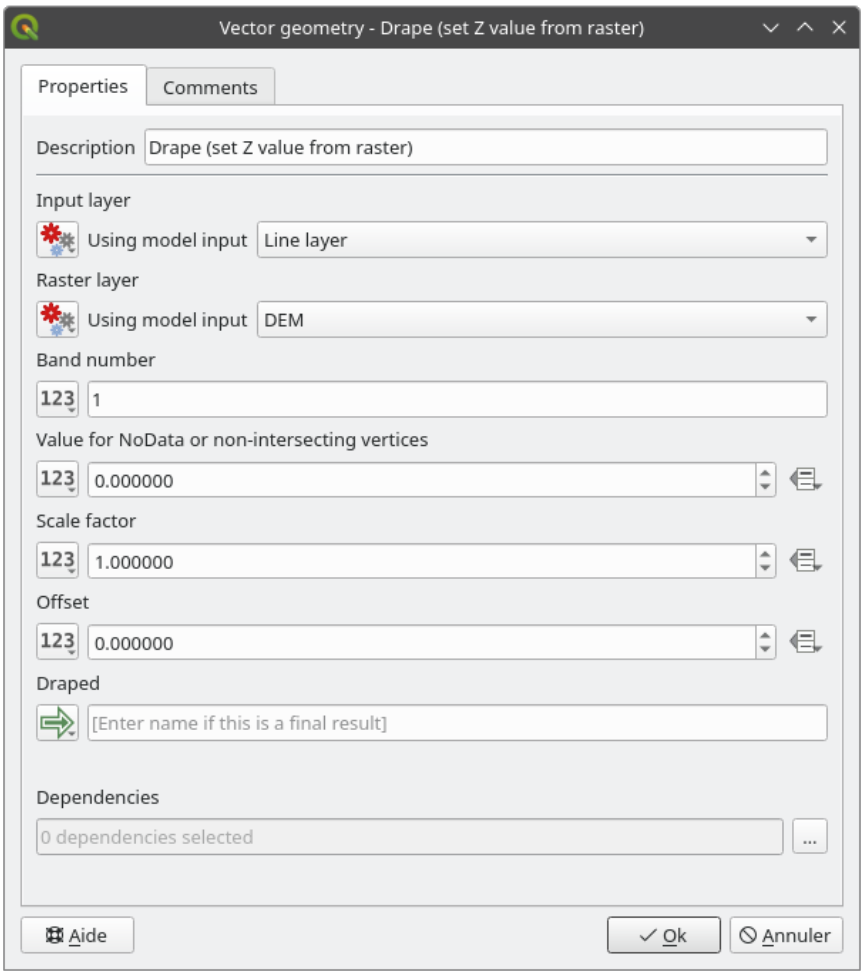


23.22 Fig. : Model Algorithm parameters

As you can see, there are however some differences. Each parameter has a drop-down menu next to it allowing to control how it will be served during the workflow:

- Value: allows you to assign a static value to the parameter. Depending on the parameter type, the widget will let you enter a number (`5.0`), a string (`mytext`), select layer(s) loaded in the QGIS project or from a folder, pick items from a list, …
- Pre-calculated Value: opens the *Expression Builder* dialog and lets you define an expression to fill the parameter. Model inputs together with some other layer statistics are available as **variables** and are listed at the top of the Search dialog of the Expression Builder. The expression is evaluated once before the child algorithm is executed and used during the execution of that algorithm.
- Model Input: allows to use an input added to the model as a parameter. Once clicked, this option will list all the suitable inputs for the parameter.
- Algorithm Output: allows to use the output of another algorithm as an input of the current algorithm. As of model inputs, this option will list all the suitable inputs for the parameter.
- The **output parameter** also has the above options in its drop-down menu:
  - add static outputs for child algorithms, e.g. always saving a child algorithm's output to a predefined geopackage or postgres layer
  - use an expression based output values for child algorithms, e.g. generating an automatic file name based on today's date and saving outputs to that file
  - use a model input, e.g. the *File/Folder* model input to specify an output file or folder
  - use another algorithm output, e.g. the output of the *Create directory* algorithm (from *Modeler tools*)
  - an addditional Model Output option makes the output of the algorithm available in the model. If a layer generated by the algorithm is only to be used as input to another algorithm, don't edit that text box.

In the following picture you can see the two input parameters defined as `Model Input` and the temporary output layer:



23.23 Fig. : Algorithm Input and Output parameters

You will also find an additional parameter named *Dependencies* that is not available when calling the algorithm from the toolbox. This parameter allows you to define the order in which algorithms are executed, by explicitly defining one algorithm as a *parent* of the current one. This will force the *parent* algorithm to be executed before the current one.

When you use the output of a previous algorithm as the input of your algorithm, that implicitly sets the previous algorithm as parent of the current one (and places the corresponding arrow in the modeler canvas). However, in some cases an algorithm might depend on another one even if it does not use any output object from it (for instance, an algorithm that executes a SQL sentence on a PostgreSQL database and another one that imports a layer into that same database). In that case, just select the previous algorithm in the *Dependencies* parameter and they will be executed in the correct order.

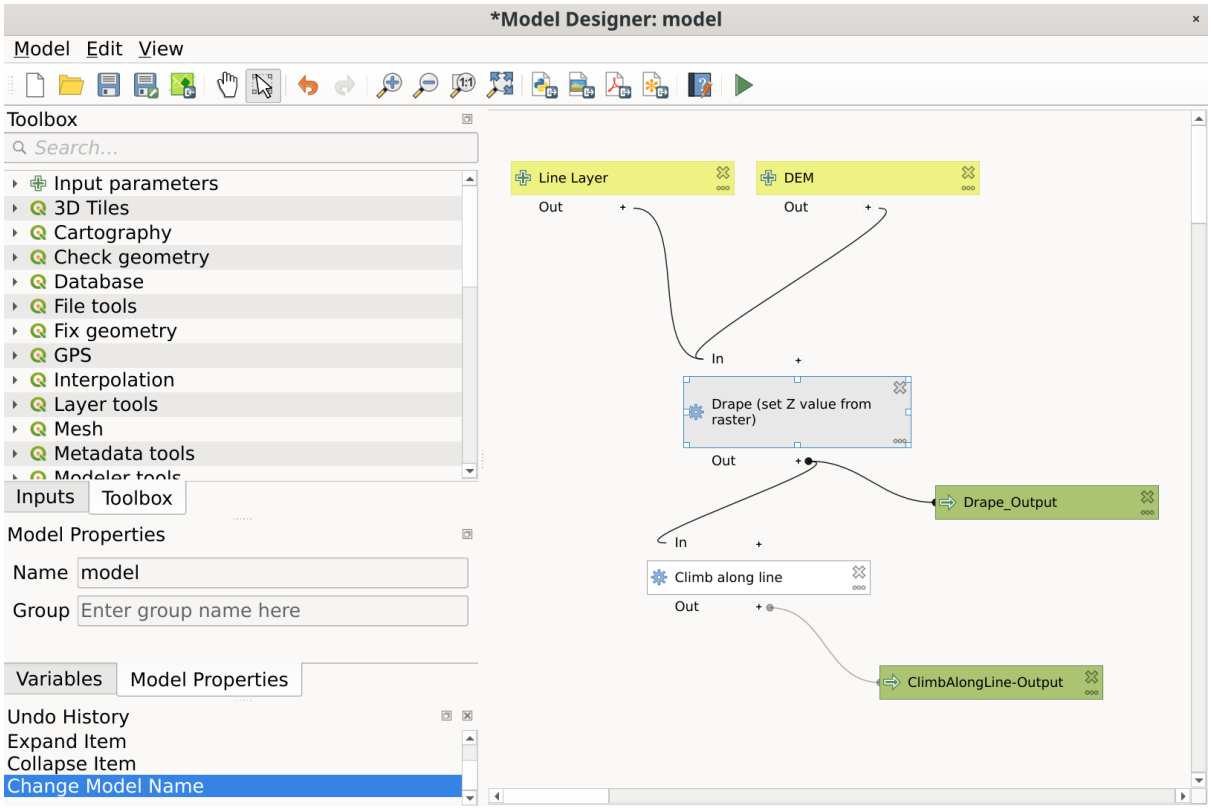
Once all the parameters have been assigned valid values, click on *OK* and the algorithm will be added to the canvas. It will be linked to the elements in the canvas (algorithms or inputs) that provide objects that are used as inputs for the algorithm.

Elements can be dragged to a different position on the canvas using the Select/Move/Link Item tool. This is useful to make the structure of the model clearer and more intuitive. You can also resize the elements, grasping their border. This is particularly useful if the description of the input or algorithm is long. With *View* ► *Enable snapping* option checked, items resizing or displacement can be bound to a virtual grid, for a more visually structured algorithm design.

Links between elements are updated automatically and you can see a + button at the top and at the bottom of each

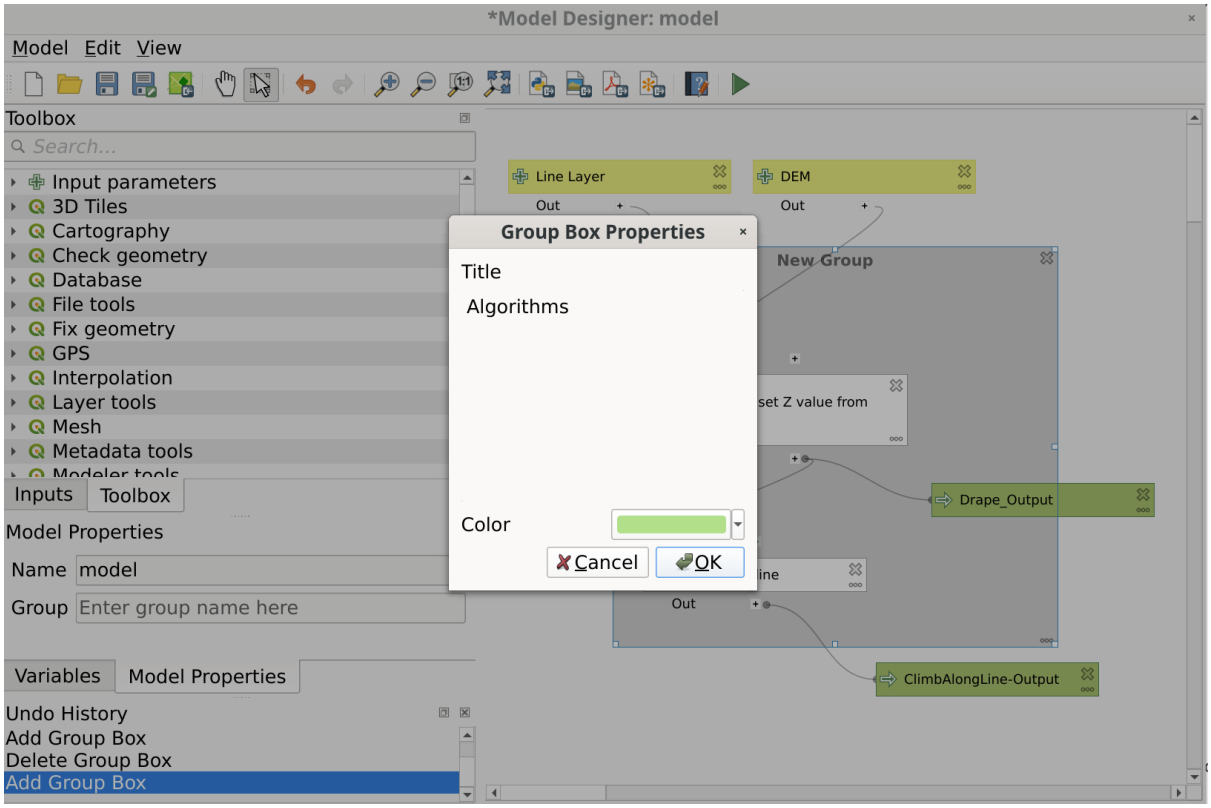
algorithm. Clicking the button will list all the inputs and outputs of the algorithm so you can have a quick overview.

You can create or remove connections between model components using an interactive drag-and-drop interface. Each parameter and algorithm item on the model canvas includes small circular sockets, output sockets on the right and input sockets on the left. To connect components, click and drag from an output socket (either from a model input or an algorithm) to an input socket on another algorithm. A visual edge follows the cursor and completes the connection when released. To remove a connection, drag the edge away from the input socket. The algorithm dialogs update automatically to reflect changes made using the drag-and-drop interface.



23.24 Fig. : A complete model

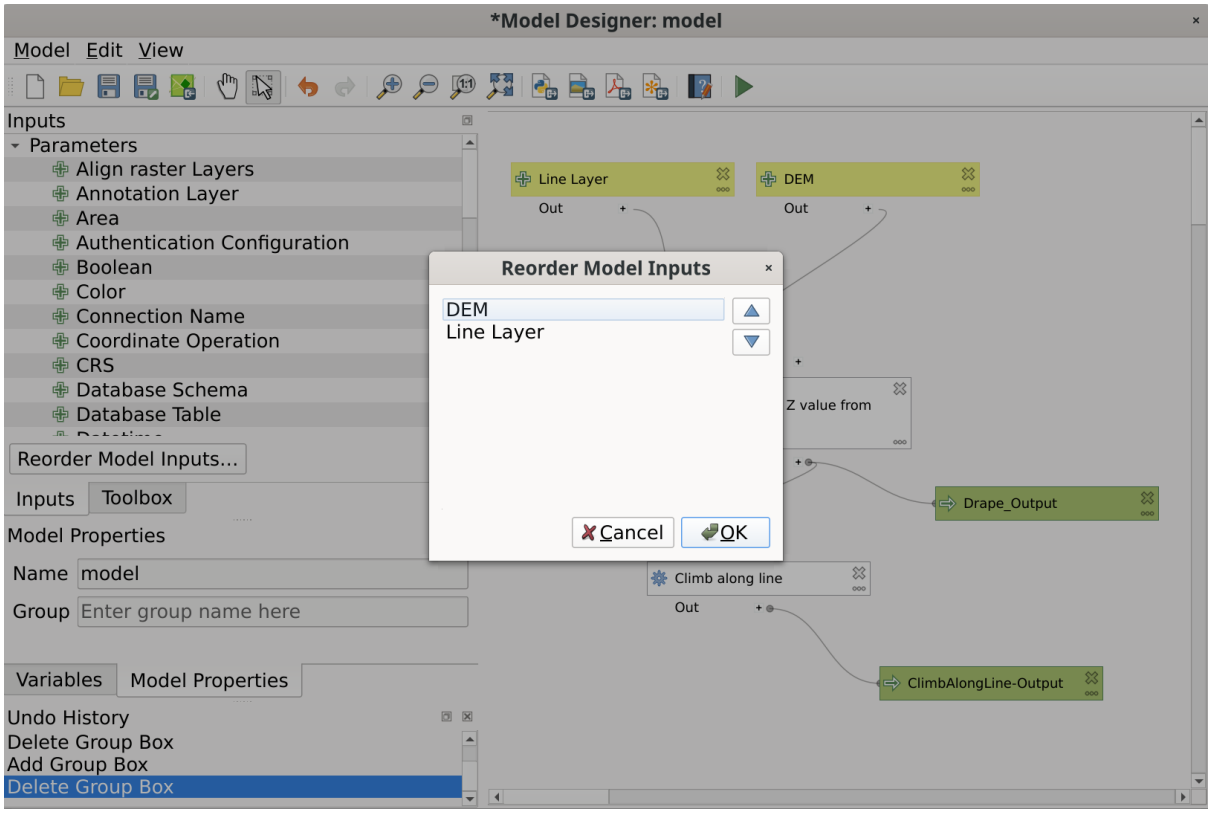
With the *Edit* ► *Add Group Box* tool, you can add a draggable *box* to the canvas. This feature is very useful in big models to group related elements in the modeler canvas and to keep the workflow clean. For example we might group together all the algorithms of the example:



23.25 Fig. : Model Group Box

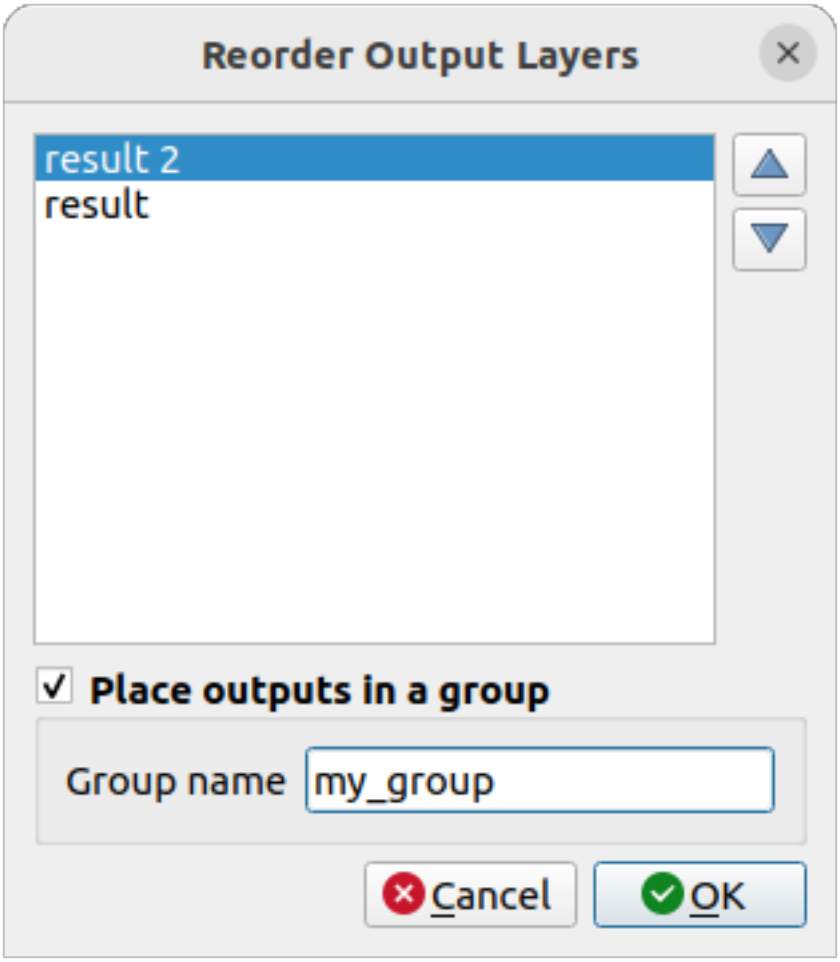
You can change the name and the color of the boxes. Group boxes are very useful when used together with *View* ► *Zoom To* ► tool, allowing you to zoom to a specific part of the model. You can also zoom in and out by using the mouse wheel.

You might want to change the order of the inputs and how they are listed in the main model dialog. At the bottom of the `Input` panel you will find the `Reorder Model Inputs...` button and by clicking on it a new dialog pops up allowing you to change the order of the inputs:



23.26 Fig. : Reorder Model Inputs

There is also the possibility to set a specific order which the outputs from the model must use when loading the results into a project. This gives the model creator a means of ensuring that layers are logically ordered on the canvas when running a model, such as placing a vector layer output over a raster layer output, or a point layer over a polygon layer. The model creator can also set an optional „Group name“ for the outputs for automatically grouping outputs within the layer tree using a new group name or by adding them to an existing group. In the `Model` menu you will find the `Reorder Output Layers...` entry and by clicking on it a new dialog pops up allowing you to change the order of the output layers:



23.27 Fig. : Reorder Output Layers

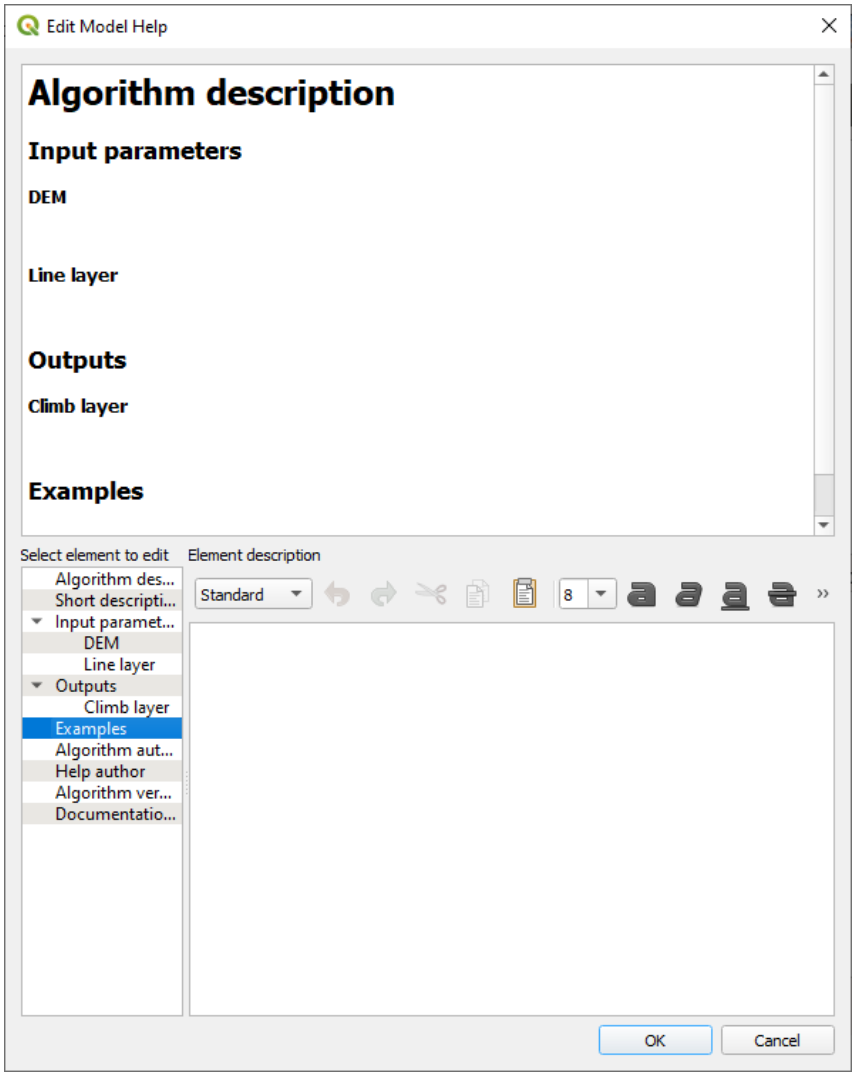
Comments can also be added to inputs or algorithms present in the modeler. This can be done by going in the *Comment* tab of the item or with a right-click. In the same tab a color can be set manual for individual model comments. Comments are visible only in the modeler canvas and not in the final algorithm dialog; they can be hidden by deactivating *View* ► *Show Comments*.

Your model can be run in various ways:

- You can run the whole model by clicking on Run Model… from the toolbar, right-clicking on the model in the *Browser panel* or *Model* ► *Run Model…*: when using the editor to execute a model, any non-default values will be saved in the inputs. This means that executing the model at a later time from the editor will have the dialog prefilled with those values on any subsequent run.
- You can select elements of the model and run only that subset of the model: press the *Run Selected Steps…* option from the *Models* ► menu or from the contextual menu of a selected algorithm. The initial state will be taken from any previous executions of the model through the editor, so results from previous steps in the model are available for the selected steps. This makes it possible to fix parts of a large model, without having to constantly run the entire model to test. Especially useful when earlier steps in the model are time consuming.
- You can run a subset of the model, starting from a specific algorithm: right-click the algorithm and select *Run from Here…*. Likewise, values are taken from previous executions.
- In order to use the algorithm from the Processing Toolbox, it has to be saved and the modeler dialog closed, to allow the toolbox to refresh its contents.

Documenting your model

You need to document your model, and this can be done from the modeler itself. Click on the Edit model help button, and a dialog like the one shown next will appear.



23.28 Fig. : Editing Help

On the right-hand side, you will see a simple HTML page, created using the description of the input parameters and outputs of the algorithm, along with some additional items like a general description of the model or its author. Also, there is an Example section where you can input your own custom examples to help explain the usage of the model. The first time you open the help editor, all these descriptions are empty, but you can edit them using the elements on the left-hand side of the dialog. Select an element on the upper part and then write its description in the text box below.

Model help is saved as part of the model itself.

23.5.3 Saving and loading models

Saving models

Use the Save model button to save the current model and the Open Model button to open a previously saved model. Models are saved with the `.model3` extension. If the model has already been saved from the modeler window, you will not be prompted for a filename. Since there is already a file associated with the model, that file will be used for subsequent saves.

Before saving a model, you have to enter a name and a group for it in the text boxes in the upper part of the window.

Models saved in the `models` folder (the default folder when you are prompted for a filename to save the model) will appear in the toolbox in the corresponding branch. When the toolbox is invoked, it searches the `models` folder for files with the `.model3` extension and loads the models they contain. Since a model is itself an algorithm, it can be added to the toolbox just like any other algorithm.

Models can also be saved within the project file using the Save model in project button. Models saved using this method won't be written as `.model3` files on the disk but will be embedded in the project file.

Project models are available in the *Project models* menu of the toolbox and in the *Project* ► *Models* menu item.

The models folder can be set from the Processing configuration dialog, under the *Modeler* group.

Models loaded from the `models` folder appear not only in the toolbox, but also in the algorithms tree in the *Algorithms* tab of the modeler window. That means that you can incorporate a model as a part of a bigger model, just like other algorithms.

Models will show up in the *Browser* panel and can be run from there.

Exporting a model as a Python script

As we will see in a later chapter, Processing algorithms can be called from the QGIS Python console, and new Processing algorithms can be created using Python. A quick way to create such a Python script is to create a model and then export it as a Python file.

To do so, click on the Export as Script Algorithm… in the modeler canvas or right click on the name of the model in the Processing Toolbox and choose Export Model as Python Algorithm….

Exporting a model as an image, PDF or SVG

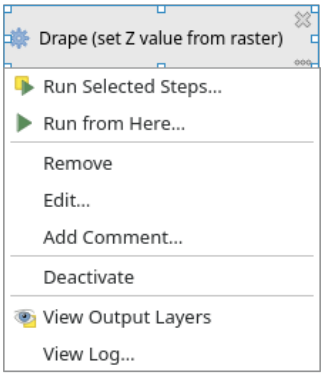
A model can also be exported as an image, SVG or PDF (for illustration purposes) by clicking Export as image, Export as PDF or Export as SVG.

23.5.4 Editing a model

You can edit the model you are currently creating, redefining the workflow and the relationships between the algorithms and inputs that define the model.

You can also edit existing model from the *Browser panel* choosing *Edit Model…* option.

If you right-click on an algorithm in the canvas, you will see a context menu like the one shown next:



23.29 Fig. : Modeler Right Click

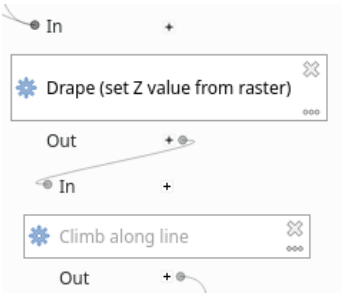
- Options for *running part of the model* have been exposed earlier.
- Selecting the *Remove* option will cause the selected algorithm to be removed. An algorithm can be removed only if there are no other algorithms depending on it. That is, if no output from the algorithm is used as input in a different one.
- Selecting the *Edit…* option will show the parameter dialog so you can change the inputs and parameter values. Not all input elements available in the model will appear as available inputs. Layers or values generated at a more advanced step in the workflow defined by the model will not be available if they cause circular dependencies.

  Select the new values and click on the *OK* button as usual. The connections between the model elements will change in the modeler canvas accordingly.
- The *Add comment…* allows you to add a comment to the algorithm to better describe the behavior.

Pastaba

Right-clicking an input parameter will also allow you to *Remove*, *Edit…* and *Add comment…*

- A model can be run partially by deactivating some of its algorithms. To do it, select the *Deactivate* option in the context menu that appears when right-clicking on an algorithm element. The selected algorithm, and all the ones in the model that depend on it will be displayed in grey and will not be executed as part of the model.



23.30 Fig. : Model with a deactivated algorithm

  When right-clicking on an algorithm that is not active, you will see an *Activate* menu option that you can use to reactivate it.
- When editing a model through the designer (and after having run that model), right-clicking any child step in the model allows to select *View Output Layers*. This will add the output layers from that step as new layers in the current QGIS project.

  This action is available for all child algorithms in the model, even if the model is not configured to use the outputs from those children as model outputs. This is designed as a helpful debugging action. If a model

fails (or gives unexpected results), you can then trace through the model and view the outputs for suspected problematic steps. It avoids the need to add temporary outputs to a model and re-run to test.

Additionally, this action is always available after running the model, even if the model itself failed (e.g., because of a misconfigured step later in the model).

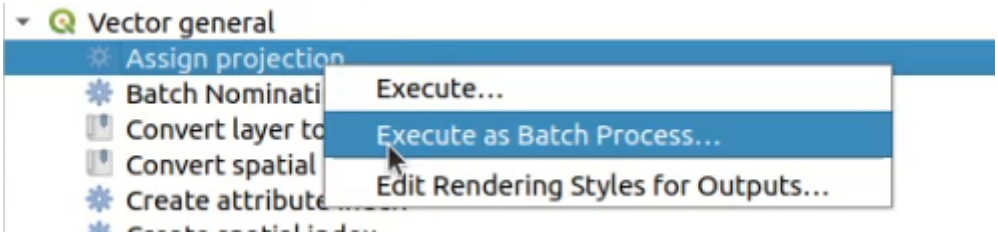
- The *View Log* action helps you see the log for each child algorithm after you've closed down the algorithm dialog.

23.6 The batch processing interface

23.6.1 Įvadas

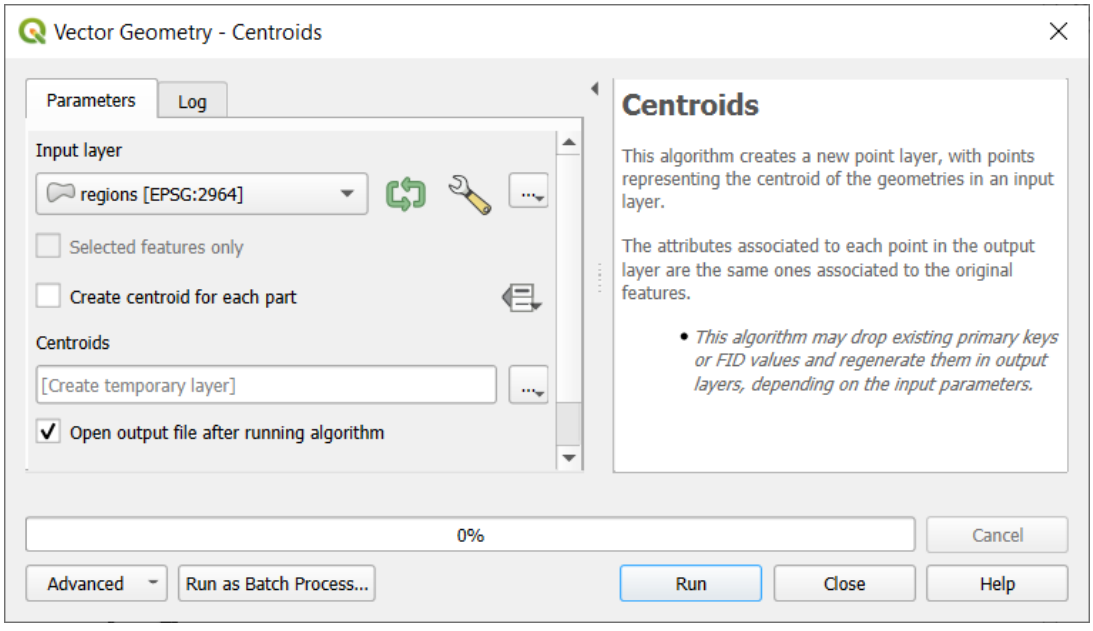
All algorithms (including models) can be executed as a batch process. That is, they can be executed using not just a single set of inputs, but several of them, executing the algorithm as many times as needed. This is useful when processing large amounts of data, since it is not necessary to launch the algorithm many times from the toolbox.

To execute an algorithm as a batch process, right-click on its name in the toolbox and select the *Execute as batch process* option in the pop-up menu that will appear.



23.31 Fig. : Batch Processing from right-click

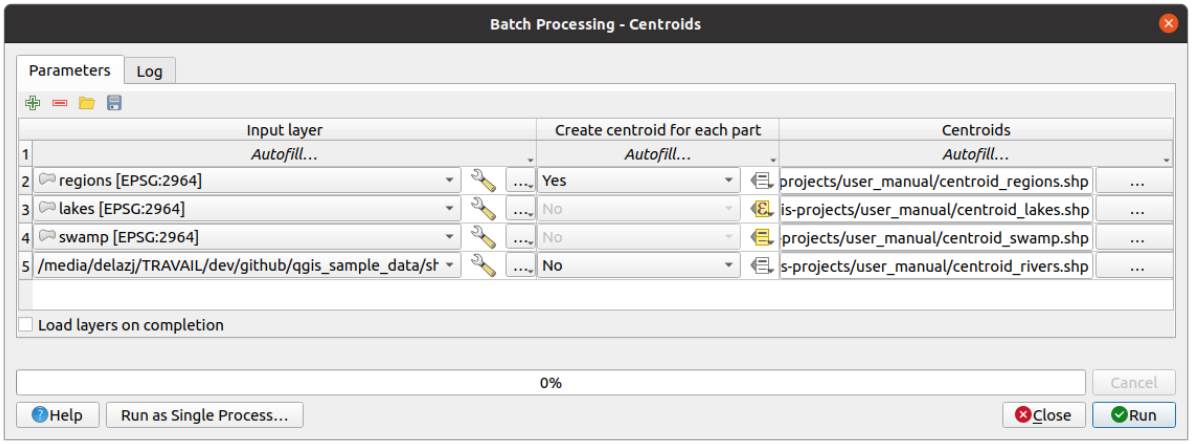
If you have the execution dialog of the algorithm open, you can also start the batch processing interface from there, clicking on the *Run as batch process...* button.



23.32 Fig. : Batch Processing From Algorithm Dialog

23.6.2 The parameters table

Executing a batch process is similar to performing a single execution of an algorithm. Parameter values have to be defined, but in this case we need not just a single value for each parameter, but a set of them instead, one for each time the algorithm has to be executed. Values are introduced using a table like the one shown next, where each row is an iteration and columns are the parameters of the algorithm.



23.33 Fig. : Batch Processing

From the top toolbar, you can:

- Toggle advanced mode: Available only when the algorithm has parameters that are marked as advanced, this button allows to show or hide such parameters in the batch dialog.
- Add row: adds a new processing entry for configuration
- Remove row(s): remove selected rows from the table. Row selection is done by clicking the number at the left and allows *keyboard combination* for multi selection.
- Open a batch processing configuration file
- Save the batch processing configuration to a `.JSON` file that can be run afterwards

By default, the table contains just two rows:

- The first row displays in each cell an *Autofill...* ► drop-down menu with *options* to quickly fill the cells below. Available options depend on the parameter type.
- The second row (as well as each subsequent one) represents a single execution of the algorithm, and each cell contains the value of one of the parameters. It is similar to the parameters dialog that you see when executing an algorithm from the toolbox, but with a different arrangement.

At the bottom of the table, you can set whether to *Load layers on completion*.

Once the size of the table has been set, it has to be filled with the desired values.

23.6.3 Filling the parameters table

For most parameters, setting the value is trivial. The appropriate widget, same as in the *single process dialog*, is provided, allowing to just type the value, or select it from a list of possible values, depending on the parameter type. This also includes data-define widget, when compatible.

To automate the batch process definition and avoid filling the table cell by cell, you may want to press down the *Autofill…* menu of a parameter and select any of the following options to replace values in the column:

- *Fill Down* will take the input for the first process and enter it for all other processes.
- *Calculate by Expression…* will allow you to create a new QGIS expression to use to update all existing values within that column. Existing parameter values (including those from other columns) are available for use inside the expression via *variables*. E.g. setting the number of segments based on the buffer distance of each layer:

```
CASE WHEN @DISTANCE > 20 THEN 12 ELSE 8 END
```

- *Add Values by Expression…* will add new rows using the values from an expression which returns an array (as opposed to *Calculate by Expression…*, which works only on existing rows). The intended use case is to allow populating the batch dialog using complex numeric series. For example adding rows for a batch buffer using the expression `generate_series(100, 1000, 50)` for distance parameter results in new rows with values 100, 150, 200, …. 1000.
- When setting a file or layer parameter, more options are provided:
  - *Add Files by Pattern…*: adds new rows to the table for files matching a *File pattern* in a folder to *Look in*. E.g. `*.shp` will add to the list all the `SHP` files in the folder. Check *Search recursively* to also browse sub-folders.
  - *Select Files…* individually on disk
  - *Add All Files from a Directory…*
  - *Select from Open Layers…* in the active project

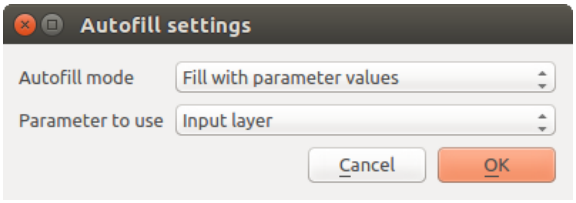
Output data parameter exposes the same capabilities as when executing the algorithm as a single process. Depending on the algorithm, the output can be:

- skipped, if the cell is left empty
- saved as a temporary layer: fill the cell with your chosen output name, select *Create Temporary Layer* from the … drop-down, and remember to tick the *Load layers on completion* checkbox. For temporary layers, the value you provide will be used as the layer name. You can use the *Autofill* options to construct that name.
- saved as a plain file (`.SHP`, `.GPKG`, `.XML`, `.PDF`, `.JPG`,…): choose *Select File/Folder…* from the … drop-down. For plain files, you can set the path using the *Autofill* options exposed beforehand.

  E.g. use *Calculate by Expression…* to set output file names to complex expressions like:

```
'/home/me/stuff/buffer_' || left(@INPUT, 30) || '_' || @DISTANCE || '.shp'
```

  You can also type the file path directly or use the file chooser dialog that appears when clicking on the accompanying *Select File/Folder…* button. Once you select the file, a new dialog is shown to allow for auto-completion of other cells in the same column (same parameter).



23.34 Fig. : Batch Processing Save

If the default value (*Do not autofill*) is selected, it will just put the selected filename in the selected cell from the parameters table. If any of the other options is selected, all the cells **below** the selected one will be automatically filled based on a defined criteria:

- *Fill with numbers*: incrementally appends a number to the file name
- *Fill with parameter values*: you can select a parameter whose value in the same row is appended to the file name. This is particularly useful for naming output data objects according to input ones.

- saved as a layer within a database container:

```
# Indicate a layer within a GeoPackage file
ogr:dbname='C:/Path/To/Geopackage.gpkg' table="New_Table" (geom)

# Use the "Calculate By Expression" to output to different layers in a␣
↪GeoPackage
'ogr:dbname=\'' || @project_folder || '/Buffers.gpkg\' table="' || @INPUT || '_
↪' || @DISTANCE || '" (geom)'
```

23.6.4 Executing the batch process

To execute the batch process once you have introduced all the necessary values, just click on *Run*. The *Log* panel is activated and displays details and steps of the execution process. Progress of the global batch task will be shown in the progress bar in the lower part of the dialog.

23.7 Using processing algorithms from the console

The console allows advanced users to increase their productivity and perform complex operations that cannot be performed using any of the other GUI elements of the processing framework. Models involving several algorithms can be defined using the command-line interface, and additional operations such as loops and conditional sentences can be added to create more flexible and powerful workflows.

There is not a processing console in QGIS, but all processing commands are available instead from the QGIS built-in *Python console*. That means that you can incorporate those commands into your console work and connect processing algorithms to all the other features (including methods from the QGIS API) available from there.

The code that you can execute from the Python console, even if it does not call any specific processing method, can be converted into a new algorithm that you can later call from the toolbox, the graphical modeler or any other component, just like you do with any other algorithm. In fact, some algorithms that you can find in the toolbox are simple scripts.

In this section, we will see how to use processing algorithms from the QGIS Python console, and also how to write algorithms using Python.

23.7.1 Calling algorithms from the Python console

The first thing you have to do is to import the processing functions with the following line:

```
>>> from qgis import processing
```

Now, there is basically just one (interesting) thing you can do with that from the console: execute an algorithm. That is done using the `run()` method, which takes the name of the algorithm to execute as its first parameter, and then a variable number of additional parameters depending on the requirements of the algorithm. So the first thing you need to know is the name of the algorithm to execute. That is not the name you see in the toolbox, but rather a unique command–line name. To find the right name for your algorithm, you can use the `processingRegistry`. Type the following line in your console:

```
>>> for alg in QgsApplication.processingRegistry().algorithms():
        print(alg.id(), "->", alg.displayName())
```

You will see something like this (with some extra dashes added to improve readability).

```
3d:tessellate --------------> Tessellate
gdal:aspect ----------------> Aspect
gdal:assignprojection ------> Assign projection
gdal:buffervectors ---------> Buffer vectors
gdal:buildvirtualraster ----> Build Virtual Raster
gdal:cliprasterbyextent ----> Clip raster by extent
gdal:cliprasterbymasklayer -> Clip raster by mask layer
gdal:clipvectorbyextent ----> Clip vector by extent
gdal:clipvectorbypolygon ---> Clip vector by mask layer
gdal:colorrelief -----------> Color relief
gdal:contour ---------------> Contour
gdal:convertformat ---------> Convert format
gdal:dissolve --------------> Dissolve
...
```

That's a list of all the available algorithm IDs, sorted by provider name and algorithm name, along with their corresponding names.

Once you know the command-line name of the algorithm, the next thing to do is to determine the right syntax to execute it. That means knowing which parameters are needed when calling the `run()` method.

There is a method to describe an algorithm in detail, which can be used to get a list of the parameters that an algorithm requires and the outputs that it will generate. To get this information, you can use the `algorithmHelp(id_of_the_algorithm)` method. Use the ID of the algorithm, not the full descriptive name.

Calling the method with `native:buffer` as parameter (`qgis:buffer` is an alias for `native:buffer` and will also work), you get the following description:

```
>>> processing.algorithmHelp("native:buffer")
Buffer (native:buffer)

This algorithm computes a buffer area for all the features in an
input layer, using a fixed or dynamic distance.

The segments parameter controls the number of line segments to
use to approximate a quarter circle when creating rounded
offsets.

The end cap style parameter controls how line endings are handled
in the buffer.

The join style parameter specifies whether round, miter or
beveled joins should be used when offsetting corners in a line.
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
The miter limit parameter is only applicable for miter join
styles, and controls the maximum distance from the offset curve
to use when creating a mitered join.


----------------
Input parameters
----------------

INPUT: Input layer

   Parameter type: QgsProcessingParameterFeatureSource

   Accepted data types:
           - str: layer ID
           - str: layer name
           - str: layer source
           - QgsProcessingFeatureSourceDefinition
           - QgsProperty
           - QgsVectorLayer

DISTANCE: Distance

   Parameter type: QgsProcessingParameterDistance

   Accepted data types:
           - int
           - float
           - QgsProperty

SEGMENTS: Segments

   Parameter type: QgsProcessingParameterNumber

   Accepted data types:
           - int
           - float
           - QgsProperty

END_CAP_STYLE: End cap style

   Parameter type: QgsProcessingParameterEnum

   Available values:
           - 0: Round
           - 1: Flat
           - 2: Square

   Accepted data types:
           - int
           - str: as string representation of int, e.g. '1'
           - QgsProperty

JOIN_STYLE: Join style

   Parameter type: QgsProcessingParameterEnum

   Available values:
           - 0: Round
           - 1: Miter
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
            - 2: Bevel

   Accepted data types:
            - int
            - str: as string representation of int, e.g. '1'
            - QgsProperty

MITER_LIMIT: Miter limit

   Parameter type: QgsProcessingParameterNumber

   Accepted data types:
            - int
            - float
            - QgsProperty

DISSOLVE: Dissolve result

   Parameter type: QgsProcessingParameterBoolean

   Accepted data types:
            - bool
            - int
            - str
            - QgsProperty

OUTPUT: Buffered

   Parameter type: QgsProcessingParameterFeatureSink

   Accepted data types:
            - str: destination vector file, e.g. 'd:/test.shp'
            - str: 'memory:' to store result in temporary memory layer
            - str: using vector provider ID prefix and destination URI,
                   e.g. 'postgres:...' to store result in PostgreSQL table
            - QgsProcessingOutputLayerDefinition
            - QgsProperty

----------------
Outputs
----------------

OUTPUT:  <QgsProcessingOutputVectorLayer>
   Buffered
```

Now you have everything you need to run any algorithm. As we have already mentioned, algorithms can be run using: `run()`. Its syntax is as follows:

```
>>> processing.run(name_of_the_algorithm, parameters)
```

Where parameters is a dictionary of parameters that depend on the algorithm you want to run, and is exactly the list that the `algorithmHelp()` method gives you.

```
>>> processing.run("native:buffer", {'INPUT': '/data/lines.shp',
               'DISTANCE': 100.0,
               'SEGMENTS': 10,
               'DISSOLVE': True,
               'END_CAP_STYLE': 0,
               'JOIN_STYLE': 0,
               'MITER_LIMIT': 10,
               'OUTPUT': '/data/buffers.shp'})
```

If a parameter is optional and you do not want to use it, then don't include it in the dictionary.

If a parameter is not specified, the default value will be used.

Depending on the type of parameter, values are introduced differently. The next list gives a quick review of how to introduce values for each type of input parameter:

- Raster Layer, Vector Layer or Table. Simply use a string with the name that identifies the data object to use (the name it has in the QGIS Table of Contents) or a filename (if the corresponding layer is not opened, it will be opened but not added to the map canvas). If you have an instance of a QGIS object representing the layer, you can also pass it as parameter.
- Enumeration. If an algorithm has an enumeration parameter, the value of that parameter should be entered using an integer value. To know the available options, you can use the `algorithmHelp()` command, as above. For instance, the `native:buffer` algorithm has an enumeration called JOIN_STYLE:

```
JOIN_STYLE: Join style

   Parameter type: QgsProcessingParameterEnum

   Available values:
           - 0: Round
           - 1: Miter
           - 2: Bevel

   Accepted data types:
           - int
           - str: as string representation of int, e.g. '1'
           - QgsProperty
```

  In this case, the parameter has three options. Notice that ordering is zero-based.

- Boolean. Use `True` or `False`.
- Multiple input. The value is a string with input descriptors separated by semicolons (`;`). As in the case of single layers or tables, each input descriptor can be the data object name, or its file path.
- Table Field from XXX. Use a string with the name of the field to use. This parameter is case-sensitive.
- Fixed Table. Type the list of all table values separated by commas (`,`) and enclosed between quotes (`"`). Values start on the upper row and go from left to right. You can also use a 2-D array of values representing the table.
- CRS. Enter the EPSG code number of the desired CRS.
- Extent. You must use a string with `xmin`, `xmax`, `ymin` and `ymax` values separated by commas (`,`).

Boolean, file, string and numerical parameters do not need any additional explanations.

Input parameters such as strings, booleans, or numerical values have default values. The default value is used if the corresponding parameter entry is missing.

For output data objects, type the file path to be used to save it, just as it is done from the toolbox. If the output object is not specified, the result is saved to a temporary file (or skipped if it is an optional output). The extension of the file determines the file format. If you enter a file extension not supported by the algorithm, the default file format for that output type will be used, and its corresponding extension appended to the given file path.

Unlike when an algorithm is executed from the toolbox, outputs are not added to the map canvas if you execute that same algorithm from the Python console using the `run()` method. That method returns a dictionary with one or more output names (the ones shown in the algorithm description) as keys and the file paths of those outputs as values:

```
>>> myresult = processing.run("native:buffer", {'INPUT': '/data/lines.shp',
              'DISTANCE': 100.0,
              'SEGMENTS': 10,
              'DISSOLVE': True,
              'END_CAP_STYLE': 0,
              'JOIN_STYLE': 0,
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
                'MITER_LIMIT': 10,
                'OUTPUT': '/data/buffers.shp'})
>>> myresult['OUTPUT']
/data/buffers.shp
```

You can then load the output in the project as any common layer:

```
>>> buffered_layer = myresult['OUTPUT']
>>> QgsProject.instance().addMapLayer(buffered_layer)
```

To immediately load the processing outputs in the project, you can use the `runAndLoadResults()` method instead of `run()`.

```
>>> processing.runAndLoadResults("native:buffer", {parameters:values})
```

If you want to open an algorithm dialog from the console you can use the `createAlgorithmDialog` method. The only mandatory parameter is the algorithm name, but you can also define the dictionary of parameters so that the dialog will be filled automatically:

```
>>> my_dialog = processing.createAlgorithmDialog("native:buffer", {
                'INPUT': '/data/lines.shp',
                'DISTANCE': 100.0,
                'SEGMENTS': 10,
                'DISSOLVE': True,
                'END_CAP_STYLE': 0,
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10,
                'OUTPUT': '/data/buffers.shp'})
>>> my_dialog.show()
```

The `execAlgorithmDialog` method opens the dialog immediately:

```
>>> processing.execAlgorithmDialog("native:buffer", {
                'INPUT': '/data/lines.shp',
                'DISTANCE': 100.0,
                'SEGMENTS': 10,
                'DISSOLVE': True,
                'END_CAP_STYLE': 0,
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10,
                'OUTPUT': '/data/buffers.shp'})
```

23.7.2 Creating scripts and running them from the toolbox

You can create your own algorithms by writing Python code. Processing scripts extend `QgsProcessingAlgorithm`, so you need to add some extra lines of code to implement mandatory functions. You can find *Create new script* (clean sheet) and *Create New Script from Template* (template that includes code for mandatory functions of `QgsProcessingAlgorithm`) under the *Scripts* dropdown menu on the top of the Processing toolbox. The Processing Script Editor will open, and that's where you should type your code. Saving the script from there in the `scripts` folder (the default folder when you open the save file dialog) with a `.py` extension should create the corresponding algorithm.

The name of the algorithm (the one you will see in the toolbox) is defined within the code.

Let's have a look at the following code, which defines a Processing algorithm that performs a buffer operation with a user defined buffer distance on a vector layer that is specified by the user, after first smoothing the layer.

```
from qgis.core import (QgsProcessingAlgorithm,
       QgsProcessingParameterNumber,
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
        QgsProcessingParameterFeatureSource,
        QgsProcessingParameterFeatureSink)

from qgis import processing

class algTest(QgsProcessingAlgorithm):
    INPUT_BUFFERDIST = 'BUFFERDIST'
    OUTPUT_BUFFER = 'OUTPUT_BUFFER'
    INPUT_VECTOR = 'INPUT_VECTOR'

    def __init__(self):
        super().__init__()

    def name(self):
        return "algTest"

    def displayName(self):
        return "algTest script"

    def createInstance(self):
        return type(self)()

    def initAlgorithm(self, config=None):
        self.addParameter(QgsProcessingParameterFeatureSource(
            self.INPUT_VECTOR, "Input vector"))
        self.addParameter(QgsProcessingParameterNumber(
            self.INPUT_BUFFERDIST, "Buffer distance",
            QgsProcessingParameterNumber.Double,
            100.0))
        self.addParameter(QgsProcessingParameterFeatureSink(
            self.OUTPUT_BUFFER, "Output buffer"))

    def processAlgorithm(self, parameters, context, feedback):
        #DO SOMETHING
        algresult = processing.run("native:smoothgeometry",
            {'INPUT': parameters[self.INPUT_VECTOR],
             'ITERATIONS':2,
             'OFFSET':0.25,
             'MAX_ANGLE':180,
             'OUTPUT': 'memory:'},
            context=context, feedback=feedback, is_child_algorithm=True)
        smoothed = algresult['OUTPUT']
        algresult = processing.run('native:buffer',
            {'INPUT': smoothed,
            'DISTANCE': parameters[self.INPUT_BUFFERDIST],
            'SEGMENTS': 5,
            'END_CAP_STYLE': 0,
            'JOIN_STYLE': 0,
            'MITER_LIMIT': 10,
            'DISSOLVE': True,
            'OUTPUT': parameters[self.OUTPUT_BUFFER]},
            context=context, feedback=feedback, is_child_algorithm=True)
        buffered = algresult['OUTPUT']
        return {self.OUTPUT_BUFFER: buffered}
```

After doing the necessary imports, the following `QgsProcessingAlgorithm` functions are specified:

- `name()`: The id of the algorithm (lowercase).
- `displayName()`: A human readable name for the algorithm.
- `createInstance()`: Create a new instance of the algorithm class.

- `initAlgorithm()`: Configure the parameterDefinitions and outputDefinitions.

  Here you describe the parameters and output of the algorithm. In this case, a feature source for the input, a feature sink for the result and a number for the buffer distance.

- `processAlgorithm()`: Do the work.

  Here we first run the `smoothgeometry` algorithm to smooth the geometry, and then we run the `buffer` algorithm on the smoothed output. To be able to run algorithms from within another algorithm we have to set the `is_child_algorithm` argument to `True`. You can see how input and output parameters are used as parameters to the `smoothgeometry` and `buffer` algorithms.

There are a number of different parameter types available for input and output. You can find the full list at *Apdorojimo algoritmų įvesties ir išvesties parametrai*.

The first parameter to the constructors is the name of the parameter, and the second is the description of the parameter (for the user interface). The rest of the constructor parameters are parameter type specific.

The input can be turned into QGIS classes using the `parameterAs` functions of `QgsProcessingAlgorithm`. For instance to get the number provided for the buffer distance as a double:

```
self.parameterAsDouble(parameters, self.INPUT_BUFFERDIST, context)).
```

The `processAlgorithm` function should return a dictionary containing values for every output defined by the algorithm. This allows access to these outputs from other algorithms, including other algorithms contained within the same model.

Well behaved algorithms should define and return as many outputs as makes sense. Non-feature outputs, such as numbers and strings, are very useful when running your algorithm as part of a larger model, as these values can be used as input parameters for subsequent algorithms within the model. Consider adding numeric outputs for things like the number of features processed, the number of invalid features encountered, the number of features output, etc. The more outputs you return, the more useful your algorithm becomes!

Feedback

The `feedback` object passed to `processAlgorithm()` should be used for user feedback / interaction. You can use the `setProgress()` function of the `feedback` object to update the progress bar (0 to 100) to inform the user about the progress of the algorithm. This is very useful if your algorithm takes a long time to complete.

The `feedback` object provides an `isCanceled()` method that should be monitored to enable cancelation of the algorithm by the user. The `pushInfo()` method of `feedback` can be used to send information to the user, and `reportError()` is handy for pushing non-fatal errors to users.

Algorithms should avoid using other forms of providing feedback to users, such as print statements or logging to `QgsMessageLog`, and should always use the feedback object instead. This allows verbose logging for the algorithm, and is also thread-safe (which is important, given that algorithms are typically run in a background thread).

Handling errors

If your algorithm encounters an error which prevents it from executing, such as invalid input values or some other condition from which it cannot or should not recover, then you should raise a `QgsProcessingException`. E.g.:

```
if feature['value'] < 20:
  raise QgsProcessingException('Invalid input value {}, must be >= 20'.
↪format(feature['value']))
```

Try to avoid raising `QgsProcessingException` for non-fatal errors (e.g. when a feature has a null geometry), and instead just report these errors via `feedback.reportError()` and skip the feature. This helps make your algorithm „model-friendly“, as it avoids halting the execution of an entire algorithm when a non-fatal error is encountered.

Documenting your scripts

As in the case of models, you can create additional documentation for your scripts, to explain what they do and how to use them.

`QgsProcessingAlgorithm` provides the `helpString()`, `shortHelpString()` and `helpUrl()` functions for that purpose. Specify / override these to provide more help to the user.

`shortDescription()` is used in the tooltip when hovering over the algorithm in the toolbox.

23.7.3 Pre- and post-execution script hooks

Scripts can also be used as pre- and post-execution hooks that are run before and after an algorithm is run, respectively. This can be used to automate tasks that should be performed whenever an algorithm is executed.

The syntax is identical to the syntax explained above, but an additional global variable named `alg` is available, representing the algorithm that has just been (or is about to be) executed.

In the *General* group of the processing options dialog, you will find two entries named *Pre-execution script* and *Post-execution script* where the filenames of the scripts to be run in each case can be entered.

23.8 Apdorojimo naudojimas komandinėje eilutėje

QGIS comes with a tool called `QGIS Processing Executor` which allows you to run Processing algorithms and models (built-in or provided by plugins) directly from the command line without starting QGIS Desktop itself.

From a command line tool, run `qgis_process` and you should get:

```
QGIS Processing Executor - 3.35.0-Master 'Master' (3.35.0-Master)
Usage: C:\OSGeo4W\apps\qgis-dev\bin\qgis_process.exe [--help] [--version] [--json]␣
↪[--verbose] [--no-python] [--skip-loading-plugins] [command] [algorithm id, path␣
↪to model file, or path to Python script] [parameters]

Options:

  --help or -h            Output the help
  --version or -v         Output all versions related to QGIS Process
  --json                  Output results as JSON objects
  --verbose               Output verbose logs
  --no-python             Disable Python support (results in faster startup)
  --skip-loading-plugins  Avoid loading enabled plugins (results in faster startup)

Available commands:

  plugins             list available and active plugins
  plugins enable      enables an installed plugin. The plugin name must be␣
↪specified, e.g. "plugins enable cartography_tools"
  plugins disable     disables an installed plugin. The plugin name must be␣
↪specified, e.g. "plugins disable cartography_tools"
  list                list all available processing algorithms
  help                show help for an algorithm. The algorithm id or a path to a␣
↪model file must be specified.
  run                 runs an algorithm. The algorithm id or a path to a model file␣
↪and parameter values must be specified.
                      Parameter values are specified after -- with PARAMETER=VALUE␣
↪syntax.
                      Ordered list values for a parameter can be created by␣
↪specifying the parameter multiple times,
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
                    e.g. --LAYERS=layer1.shp --LAYERS=layer2.shp
                    Alternatively, a '-' character in place of the parameters␣
↪argument indicates that the parameters should be read from STDIN as a JSON␣
↪object.
                    The JSON should be structured as a map containing at least␣
↪the "inputs" key specifying a map of input parameter values.
                    This implies the --json option for output as a JSON object.
                    If required, the ellipsoid to use for distance and area␣
↪calculations can be specified via the "--ELLIPSOID=name" argument.
                    If required, an existing QGIS project to use during the␣
↪algorithm execution can be specified via the "--PROJECT_PATH=path" argument.
                    When passing parameters as a JSON object from STDIN, these␣
↪extra arguments can be provided as an "ellipsoid" and a "project_path" key␣
↪respectively.
```

Pastaba

Only installed plugins that advertise `hasProcessingProvider=yes` in their `metadata.txt` file are recognized and can be activated or loaded by qgis_process tool.

Patarimas

Before calling qgis_process on a system without window manager (e.g. a headless server), you should set:

```
export QT_QPA_PLATFORM=offscreen
```

The command `list` can be used to get a list of all available providers and algorithms.

```
qgis_process list
```

The command `help` can be used to get further information about commands or algorithms.

```
qgis_process help qgis:regularpoints
```

The command `run` can be used to run an algorithm or model. Specify the name of the algorithm or a path to a model as first parameter.

```
qgis_process run native:buffer -- INPUT=source.shp DISTANCE=2 OUTPUT=buffered.shp
```

Where a parameter accepts a list of values, set the same variable multiple times.

```
qgis_process run native:mergevectorlayers -- LAYERS=input1.shp LAYERS=input2.shp␣
↪OUTPUT=merged.shp
```

While running an algorithm a text-based feedback bar is shown, and the operation can be cancelled via `CTRL+C`.

The `run` command also supports further parameters.

- `--json` will format stdout output in a JSON structured way.
- `--ellipsoid` will set the ellipsoid to the specified one.
- `--distance_units` will use the specified distance units.
- `--area_units` will use the specified area units.
- `--project_path` will load the specified project for running the algorithm.

Complex input parameters, i.e. parameter types which are themselves specified as a dictionary type object for algorithms, are supported by qgis_process. To indicate that parameters will be specified via stdin, the qgis_process command must follow the format (with a trailing – in place of the usual arguments list).

```
qgis_process run algorithmId –
```

The JSON object must contain an „inputs“ key, which is a map of the input parameter values. E.g.

```
echo "{'inputs': {'INPUT': 'my_shape.shp', 'DISTANCE': 5}}" | qgis_process run␣
↪native:buffer –
```

Additionally, extra settings like the distance units, area units, ellipsoid and project path can be included in this JSON object:

```
{
 'ellipsoid': 'EPSG:7019',
 'distance_units': 'feet',
 'area_units': 'ha',
 'project_path': 'C:/temp/my_project.qgs'
 'inputs': {'DISTANCE': 5, 'SEGMENTS': 8 ... }
}
```

Specifying input parameters via stdin implies automatically the `JSON` output format for results.

23.9 Naujų apdorojimo algoritmų rašymas Python scenarijais

There are two options for writing Processing algorithms using Python.

- *Extending* `QgsProcessingAlgorithm`
- *Using the @alg decorator*

Within QGIS, you can use *Create new script* in the *Scripts* menu at the top of the *Processing Toolbox* to open the *Processing Script Editor* where you can write your code. To simplify the task, you can start with a script template by using *Create new script from template* from the same menu. This opens a template that extends `QgsProcessingAlgorithm`.

If you save the script in the `scripts` folder (the default location) with a `.py` extension, the algorithm will become available in the *Processing Toolbox*.

23.9.1 Extending QgsProcessingAlgorithm

The following code

1. takes a vector layer as input
2. counts the number of features
3. does a buffer operation
4. creates a raster layer from the result of the buffer operation
5. returns the buffer layer, raster layer and number of features

```
1 from qgis.PyQt.QtCore import QCoreApplication
2 from qgis.core import (QgsProcessing,
3                        QgsProcessingAlgorithm,
4                        QgsProcessingException,
5                        QgsProcessingOutputNumber,
6                        QgsProcessingParameterDistance,
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
                        QgsProcessingParameterFeatureSource,
                        QgsProcessingParameterVectorDestination,
                        QgsProcessingParameterRasterDestination)
from qgis import processing


class ExampleProcessingAlgorithm(QgsProcessingAlgorithm):
    """
    This is an example algorithm that takes a vector layer,
    creates some new layers and returns some results.
    """

    def tr(self, string):
        """
        Returns a translatable string with the self.tr() function.
        """
        return QCoreApplication.translate('Processing', string)

    def createInstance(self):
        # Must return a new copy of your algorithm.
        return ExampleProcessingAlgorithm()

    def name(self):
        """
        Returns the unique algorithm name.
        """
        return 'bufferrasterextend'

    def displayName(self):
        """
        Returns the translated algorithm name.
        """
        return self.tr('Buffer and export to raster (extend)')

    def group(self):
        """
        Returns the name of the group this algorithm belongs to.
        """
        return self.tr('Example scripts')

    def groupId(self):
        """
        Returns the unique ID of the group this algorithm belongs
        to.
        """
        return 'examplescripts'

    def shortHelpString(self):
        """
        Returns a localised short help string for the algorithm.
        """
        return self.tr('Example algorithm short description')

    def initAlgorithm(self, config=None):
        """
        Here we define the inputs and outputs of the algorithm.
        """
        # 'INPUT' is the recommended name for the main input
        # parameter.
        self.addParameter(
            QgsProcessingParameterFeatureSource(
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
                'INPUT',
                self.tr('Input vector layer'),
                types=[QgsProcessing.TypeVectorAnyGeometry]
            )
        )
        self.addParameter(
            QgsProcessingParameterVectorDestination(
                'BUFFER_OUTPUT',
                self.tr('Buffer output'),
            )
        )
        # 'OUTPUT' is the recommended name for the main output
        # parameter.
        self.addParameter(
            QgsProcessingParameterRasterDestination(
                'OUTPUT',
                self.tr('Raster output')
            )
        )
        self.addParameter(
            QgsProcessingParameterDistance(
                'BUFFERDIST',
                self.tr('BUFFERDIST'),
                defaultValue = 1.0,
                # Make distance units match the INPUT layer units:
                parentParameterName='INPUT'
            )
        )
        self.addParameter(
            QgsProcessingParameterDistance(
                'CELLSIZE',
                self.tr('CELLSIZE'),
                defaultValue = 10.0,
                parentParameterName='INPUT'
            )
        )
        self.addOutput(
            QgsProcessingOutputNumber(
                'NUMBEROFFEATURES',
                self.tr('Number of features processed')
            )
        )

    def processAlgorithm(self, parameters, context, feedback):
        """
        Here is where the processing itself takes place.
        """
        # First, we get the count of features from the INPUT layer.
        # This layer is defined as a QgsProcessingParameterFeatureSource
        # parameter, so it is retrieved by calling
        # self.parameterAsSource.
        input_featuresource = self.parameterAsSource(parameters,
                                                     'INPUT',
                                                     context)
        numfeatures = input_featuresource.featureCount()

        # Retrieve the buffer distance and raster cell size numeric
        # values. Since these are numeric values, they are retrieved
        # using self.parameterAsDouble.
        bufferdist = self.parameterAsDouble(parameters, 'BUFFERDIST',
                                            context)
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
        rastercellsize = self.parameterAsDouble(parameters, 'CELLSIZE',
                                                context)
        if feedback.isCanceled():
            return {}
        buffer_result = processing.run(
            'native:buffer',
            {
                # Here we pass on the original parameter values of INPUT
                # and BUFFER_OUTPUT to the buffer algorithm.
                'INPUT': parameters['INPUT'],
                'OUTPUT': parameters['BUFFER_OUTPUT'],
                'DISTANCE': bufferdist,
                'SEGMENTS': 10,
                'DISSOLVE': True,
                'END_CAP_STYLE': 0,
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10
            },
            # Because the buffer algorithm is being run as a step in
            # another larger algorithm, the is_child_algorithm option
            # should be set to True
            is_child_algorithm=True,
            #
            # It's important to pass on the context and feedback objects to
            # child algorithms, so that they can properly give feedback to
            # users and handle cancelation requests.
            context=context,
            feedback=feedback)

        # Check for cancelation
        if feedback.isCanceled():
            return {}

        # Run the separate rasterization algorithm using the buffer result
        # as an input.
        rasterized_result = processing.run(
            'qgis:rasterize',
            {
                # Here we pass the 'OUTPUT' value from the buffer's result
                # dictionary off to the rasterize child algorithm.
                'LAYER': buffer_result['OUTPUT'],
                'EXTENT': buffer_result['OUTPUT'],
                'MAP_UNITS_PER_PIXEL': rastercellsize,
                # Use the original parameter value.
                'OUTPUT': parameters['OUTPUT']
            },
            is_child_algorithm=True,
            context=context,
            feedback=feedback)

        if feedback.isCanceled():
            return {}

        # Return the results
        return {'OUTPUT': rasterized_result['OUTPUT'],
                'BUFFER_OUTPUT': buffer_result['OUTPUT'],
                'NUMBEROFFEATURES': numfeatures}
```

Processing algorithm standard functions:

- **createInstance (mandatory)**
  Must return a new copy of your algorithm. If you change the name of the class, make sure you also

update the value returned here to match!

- **name (mandatory)**
  Returns the unique algorithm name, used for identifying the algorithm.
- **displayName (mandatory)**
  Returns the translated algorithm name.
- **group**
  Returns the name of the group this algorithm belongs to.
- **groupId**
  Returns the unique ID of the group this algorithm belongs to.
- **shortHelpString**
  Returns a localised short help string for the algorithm.
- **initAlgorithm (mandatory)**
  Here we define the inputs and outputs of the algorithm.

  `INPUT` and `OUTPUT` are recommended names for the main input and main output parameters, respectively.

  If a parameter depends on another parameter, `parentParameterName` is used to specify this relationship (could be the field / band of a layer or the distance units of a layer).
- **processAlgorithm (mandatory)**
  This is where the processing takes place.

  Parameters are retrieved using special purpose functions, for instance `parameterAsSource` and `parameterAsDouble`.

  `processing.run` can be used to run other processing algorithms from a processing algorithm. The first parameter is the name of the algorithm, the second is a dictionary of the parameters to the algorithm. `is_child_algorithm` is normally set to `True` when running an algorithm from within another algorithm. `context` and `feedback` inform the algorithm about the environment to run in and the channel for communicating with the user (catching cancel request, reporting progress, providing textual feedback). When using the (parent) algorithm's parameters as parameters to „child" algorithms, the original parameter values should be used (e.g. `parameters['OUTPUT']`).

  It is good practice to check the feedback object for cancelation as much as is sensibly possible! Doing so allows for responsive cancelation, instead of forcing users to wait for unwanted processing to occur.

  The algorithm should return values for all the output parameters it has defined as a dictionary. In this case, that's the buffer and rasterized output layers, and the count of features processed. The dictionary keys must match the original parameter/output names.

23.9.2 The @alg decorator

Using the @alg decorator, you can create your own algorithms by writing the Python code and adding a few extra lines to supply additional information needed to make it a proper Processing algorithm. This simplifies the creation of algorithms and the specification of inputs and outputs.

One important limitation with the decorator approach is that algorithms created in this way will always be added to a user's Processing Scripts provider – it is not possible to add these algorithms to a custom provider, e.g. for use in plugins.

The following code uses the @alg decorator to

1. use a vector layer as input
2. count the number of features
3. do a buffer operation
4. create a raster layer from the result of the buffer operation

5. returns the buffer layer, raster layer and number of features

```
from qgis import processing
from qgis.processing import alg
from qgis.core import QgsProject

@alg(name='bufferrasteralg', label='Buffer and export to raster (alg)',
     group='examplescripts', group_label='Example scripts')
# 'INPUT' is the recommended name for the main input parameter
@alg.input(type=alg.SOURCE, name='INPUT', label='Input vector layer')
# 'OUTPUT' is the recommended name for the main output parameter
@alg.input(type=alg.RASTER_LAYER_DEST, name='OUTPUT',
           label='Raster output')
@alg.input(type=alg.VECTOR_LAYER_DEST, name='BUFFER_OUTPUT',
           label='Buffer output')
@alg.input(type=alg.DISTANCE, name='BUFFERDIST', label='BUFFER DISTANCE',
           default=1.0)
@alg.input(type=alg.DISTANCE, name='CELLSIZE', label='RASTER CELL SIZE',
           default=10.0)
@alg.output(type=alg.NUMBER, name='NUMBEROFFEATURES',
            label='Number of features processed')

def bufferrasteralg(instance, parameters, context, feedback, inputs):
    """
    Description of the algorithm.
    (If there is no comment here, you will get an error)
    """
    input_featuresource = instance.parameterAsSource(parameters,
                                                     'INPUT', context)
    numfeatures = input_featuresource.featureCount()
    bufferdist = instance.parameterAsDouble(parameters, 'BUFFERDIST',
                                            context)
    rastercellsize = instance.parameterAsDouble(parameters, 'CELLSIZE',
                                                context)
    if feedback.isCanceled():
        return {}
    buffer_result = processing.run('native:buffer',
                                   {'INPUT': parameters['INPUT'],
                                    'OUTPUT': parameters['BUFFER_OUTPUT'],
                                    'DISTANCE': bufferdist,
                                    'SEGMENTS': 10,
                                    'DISSOLVE': True,
                                    'END_CAP_STYLE': 0,
                                    'JOIN_STYLE': 0,
                                    'MITER_LIMIT': 10
                                    },
                                   is_child_algorithm=True,
                                   context=context,
                                   feedback=feedback)
    if feedback.isCanceled():
        return {}
    rasterized_result = processing.run('qgis:rasterize',
                                   {'LAYER': buffer_result['OUTPUT'],
                                    'EXTENT': buffer_result['OUTPUT'],
                                    'MAP_UNITS_PER_PIXEL': rastercellsize,
                                    'OUTPUT': parameters['OUTPUT']
                                   },
                                   is_child_algorithm=True, context=context,
                                   feedback=feedback)
    if feedback.isCanceled():
        return {}
    return {'OUTPUT': rasterized_result['OUTPUT'],
            'BUFFER_OUTPUT': buffer_result['OUTPUT'],
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
62              'NUMBEROFFEATURES': numfeatures}
```

As you can see, it involves two algorithms (‚native:buffer' and ‚qgis:rasterize'). The last one (‚qgis:rasterize') creates a raster layer from the buffer layer that was generated by the first one (‚native:buffer').

The part of the code where this processing takes place is not difficult to understand if you have read the previous chapter. The first lines, however, need some additional explanation. They provide the information that is needed to turn your code into an algorithm that can be run from any of the GUI components, like the toolbox or the model designer.

These lines are all calls to the `@alg` decorator functions that help simplify the coding of the algorithm.

- The @alg decorator is used to define the name and location of the algorithm in the Toolbox.
- The @alg.input decorator is used to define the inputs of the algorithm.
- The @alg.output decorator is used to define the outputs of the algorithm.

For existing parameters and their correspondance, read *Apdorojimo algoritmų įvesties ir išvesties parametrai*.

23.9.3 Handing algorithm output

When you declare an output representing a layer (raster or vector), the algorithm will try to add it to QGIS once it is finished.

- Raster layer output: QgsProcessingParameterRasterDestination / alg.RASTER_LAYER_DEST.
- Vector layer output: QgsProcessingParameterVectorDestination / alg.VECTOR_LAYER_DEST.

So even if the `processing.run()` method does not add the layers it creates to the user's current project, the two output layers (buffer and raster buffer) will be loaded, since they are saved to the destinations entered by the user (or to temporary destinations if the user does not specify destinations).

If a layer is created as output of an algorithm, it should be declared as such. Otherwise, you will not be able to properly use the algorithm in the modeler, since what is declared will not match what the algorithm really creates.

You can return strings, numbers and more by specifying them in the result dictionary (as demonstrated for „NUMBEROFFEATURES"), but they should always be explicitly defined as outputs from your algorithm. We encourage algorithms to output as many useful values as possible, since these can be valuable for use in later algorithms when your algorithm is used as part of a model.

23.9.4 Communicating with the user

If your algorithm takes a long time to process, it is a good idea to inform the user about the progress. You can use `feedback` (`QgsProcessingFeedback`) for this.

The progress text and progressbar can be updated using two methods: `setProgressText(text)` and `setProgress(percent)`.

You can provide more information by using `pushCommandInfo(text)`, `pushDebugInfo(text)`, `pushInfo(text)` and `reportError(text)`.

If your script has a problem, the correct way of handling it is to raise a `QgsProcessingException`. You can pass a message as an argument to the constructor of the exception. Processing will take care of handling it and communicating with the user, depending on where the algorithm is being executed from (toolbox, modeler, Python console, …)

23.9.5 Documenting your scripts

You can document your scripts by overloading the `helpString()` and `helpUrl()` methods of `QgsProcessingAlgorithm`.

23.9.6 Flags

You can override the `flags()` method of `QgsProcessingAlgorithm` to tell QGIS more about your algorithm. You can for instance tell QGIS that the script shall be hidden from the modeler, that it can be canceled, that it is not thread safe, and more.

Patarimas

By default, Processing runs algorithms in a separate thread in order to keep QGIS responsive while the processing task runs. If your algorithm is regularly crashing, you are probably using API calls which are not safe to do in a background thread. Try returning the QgsProcessingAlgorithm.FlagNoThreading flag from your algorithm's flags() method to force Processing to run your algorithm in the main thread instead.

23.9.7 Best practices for writing script algorithms

Here's a quick summary of ideas to consider when creating your script algorithms and, especially, if you want to share them with other QGIS users. Following these simple rules will ensure consistency across the different Processing elements such as the toolbox, the modeler or the batch processing interface.

- Do not load resulting layers. Let Processing handle your results and load your layers if needed.
- Always declare the outputs your algorithm creates.
- Do not show message boxes or use any GUI element from the script. If you want to communicate with the user, use the methods of the feedback object (`QgsProcessingFeedback`) or throw a `QgsProcessingException`.

There are already many processing algorithms available in QGIS. You can find code on the QGIS repo.

23.10 Apdorojimo algoritmų įvesties ir išvesties parametrai

Here is the list of input and output types that are supported in Processing with their corresponding alg decorator constants, their name in the modeler designer if exposed.

23.10.1 Input types

Input name	Class	Alg constant	Aprašymas
Annotation layer	`QgsProcessingParame`	`alg.ANNOTATION_LAYE`	An annotation layer
Area	`QgsProcessingParame`		A numeric parameter representing an area measure
Authentication Configuration	`QgsProcessingParame`	`alg.AUTH_CFG`	Allows users to select from available authentication configurations or create new authentication configurations

continues on next page

Table 23.1 – tęsinys iš praeito puslapio

Input name	Class	Alg constant	Aprašymas
Raster band	QgsProcessingParame	alg.BAND	A band of a raster layer
Boolean	QgsProcessingParame	alg.BOOL	A boolean value
Connection Name	QgsProcessingParame		A selection from available registered connections for a particular data provider
Color	QgsProcessingParame	alg.COLOR	A color
Coordinate Operation	QgsProcessingParame	alg.COORDINATE_OPER	A coordinate operation (for CRS transformations)
CRS	QgsProcessingParame	alg.CRS	A Coordinate Reference System
Database Schema	QgsProcessingParame	alg.DATABASE_SCHEMA	A database schema
Database Table	QgsProcessingParame	alg.DATABASE_TABLE	A database table
Data	ProcessingDateTimeP Date	alg.DATE	A pure date
Datetime	ProcessingDateTimeP DateTime	alg.DATETIME	A datetime
Distance	QgsProcessingParame	alg.DISTANCE	A double numeric parameter for distance values
DXF Layers	QgsProcessingParame		A list of vector layers to export as DXF file
Enum	QgsProcessingParame	alg.ENUM	An enumeration, allowing for selection from a set of predefined values
Expression	QgsProcessingParame	alg.EXPRESSION	An expression
Extent	QgsProcessingParame	alg.EXTENT	A spatial extent defined by xmin, xmax, ymin, ymax
	QgsProcessingParame	alg.FIELD	A field in the attribute table of a vector layer
Field Aggregates	QgsProcessingParame		A parameter for “aggregate” configurations, which consist of a definition of desired output fields, types, and aggregate used to populate them
Fields Mapper	QgsProcessingParame		A parameter for “field mapping” configurations, which consist of a definition of desired output fields, types, and expressions used to populate them
File/Folder	QgsProcessingParame	alg.FILE	A filename of an existing file
	QgsProcessingParame	alg.FILE_DEST	A filename for a newly created output file
	QgsProcessingParame	alg.FOLDER_DEST	A folder (destination folder)
Geometrija	QgsProcessingParame	alg.GEOMETRY	A geometry
	ProcessingNumberPar Integer	alg.INT	An integer
Spausdinimo išdėstymas	QgsProcessingParame	alg.LAYOUT	A layout
Print Layout Item	QgsProcessingParame	alg.LAYOUT_ITEM	A layout item
Map Layer	QgsProcessingParame	alg.MAPLAYER	A map layer
Map Theme	QgsProcessingParame	alg.MAP_THEME	A project map theme
Matrix	QgsProcessingParame	alg.MATRIX	A matrix

continues on next page

Table 23.1 – tęsinys iš praeito puslapio

Input name	Class	Alg constant	Aprašymas
Mesh Dataset Groups	QgsProcessingParame		Dataset groups of mesh layer
Mesh Dataset Time	QgsProcessingParame		Dataset index from a mesh layer by time setting
Tinklelio sluoksnis	QgsProcessingParame	alg. MESH_LAYER	A mesh layer
Multiple Input	QgsProcessingParame	alg. MULTILAYER	A set of layers
Number	ProcessingNumberPar Double	alg.NUMBER	A numerical value
Taškai	QgsProcessingParame	alg.POINT	A point
Point Cloud Attribute	QgsProcessingParame		An attribute from a point cloud layer
	QgsProcessingParame	alg. POINTCLOUD_LAYE	A point cloud layer destination parameter, for specifying the destination path for a point cloud layer created by the algorithm
Taškų debesies sluoksnis	QgsProcessingParame	alg. POINTCLOUD_LAYE	A point cloud layer
	QgsProcessingParame	alg. PROVIDER_CONNEC	An available connection for a database provider
Diapazonas	QgsProcessingParame	alg.RANGE	A number range
Rastro sluoksnis	QgsProcessingParame	alg. RASTER_LAYER	A raster layer
	QgsProcessingParame	alg. RASTER_LAYER_DE	A raster layer destination parameter, for specifying the destination path for a raster layer created by the algorithm
Scale	QgsProcessingParame	alg.SCALE	A map scale
	QgsProcessingParame	alg.SINK	A feature sink
Tekstas	QgsProcessingParame	alg.STRING	A text string
TIN Creation Layers	QgsProcessingParame		Selection of multiple layers to create a TIN with vertices and/or break lines
Time	ProcessingDateTimeP Time	alg.TIME	A pure time
Vector Features	QgsProcessingParame	alg.SOURCE	A feature source
Vektorinis sluoksnis	QgsProcessingParame	alg. VECTOR_LAYER	A vector layer
	QgsProcessingParame	alg. VECTOR_LAYER_DE	A vector layer destination parameter, for specifying the destination path for a vector layer created by the algorithm
Vector Tile Writer Layers	QgsProcessingParame		A vector tile layer destination parameter, for specifying the destination path for a vector tile layer created by the algorithm
Volume	QgsProcessingParame		A numeric parameter representing a volume measure

23.10.2 Output types

Class	Alg constant	Aprašymas
QgsProcessingOutputBoolean	alg.BOOL	A boolean value
QgsProcessingOutputNumber	alg.DISTANCE	A double numeric parameter for distance values
QgsProcessingOutputFile	alg.FILE	A filename of an existing file
QgsProcessingOutputFolder	alg.FOLDER	A folder
QgsProcessingOutputHtml	alg.HTML	HTML
QgsProcessingOutputNumber	alg.INT	A integer
QgsProcessingOutputLayerDefinition	alg.LAYERDEF	A layer definition
QgsProcessingOutputMapLayer	alg.MAPLAYER	A map layer
QgsProcessingOutputMultipleLayers	alg.MULTILAYER	A set of layers
QgsProcessingOutputNumber	alg.NUMBER	A numerical value
QgsProcessingOutputPointCloudLayer	alg.POINTCLOUD_LAYER	A point cloud layer
QgsProcessingOutputRasterLayer	alg.RASTER_LAYER	A raster layer
QgsProcessingOutputString	alg.STRING	A text string
QgsProcessingOutputVectorLayer	alg.VECTOR_LAYER	A vector layer
QgsProcessingOutputVectorTileLayer		A vector tile layer

23.11 Išorinių aplikacijų konfigūravimas

The Processing framework can be extended using additional applications. Algorithms that rely on external applications are managed by their own algorithm providers. Additional providers can be found as separate plugins, and installed using the QGIS Plugin Manager.

This section will show you how to configure the Processing framework to include these additional applications, and it will explain some particular features of the algorithms based on them. Once you have correctly configured the system, you will be able to execute external algorithms from any component like the toolbox or the model designer, just like you do with any other algorithm.

By default, algorithms that rely on an external application not shipped with QGIS are not enabled. You can enable them in the Processing settings dialog if they are installed on your system.

23.11.1 A note for Windows users

If you are not an advanced user and you are running QGIS on Windows, you might not be interested in reading the rest of this chapter. Make sure you install QGIS in your system using the standalone installer. That will automatically install GRASS in your system and configure it so it can be run from QGIS. All the algorithms from this provider will be ready to be run without needing any further configuration. If installing with the OSGeo4W application, make sure that you also select GRASS for installation.

23.11.2 A note on file formats

When using external software, opening a file in QGIS does not mean that it can be opened and processed in that other software. In most cases, other software can read what you have opened in QGIS, but in some cases, that might not be true. When using databases or uncommon file formats, whether for raster or vector layers, problems might arise. If that happens, try to use well-known file formats that you are sure are understood by both programs, and check the console output (in the log panel) to find out what is going wrong.

You might for instance get trouble and not be able to complete your work if you call an external algorithm with a GRASS raster layers as input. For this reason, such layers will not appear as available to algorithms.

You should, however, not have problems with vector layers, since QGIS automatically converts from the original file format to one accepted by the external application before passing the layer to it. This adds extra processing time, which might be significant for large layers, so do not be surprised if it takes more time to process a layer from a DB connection than a layer from a Shapefile format dataset of similar size.

Providers not using external applications can process any layer that you can open in QGIS, since they open it for analysis through QGIS.

All raster and vector output formats produced by QGIS can be used as input layers. Some providers do not support certain formats, but all can export to common formats that can later be transformed by QGIS automatically. As for input layers, if a conversion is needed, that might increase the processing time.

23.11.3 A note on vector layer selections

External applications may also be made aware of the selections that exist in vector layers within QGIS. However, that requires rewriting all input vector layers, just as if they were originally in a format not supported by the external application. Only when no selection exists, or the *Use only selected features* option is not enabled in the processing general configuration, can a layer be directly passed to an external application.

In other cases, exporting only selected features is needed, which causes longer execution times.

23.11.4 Using third-party Providers

R scripts and libraries

To enable R in Processing you need to install the **Processing R Provider** plugin and configure R for QGIS.

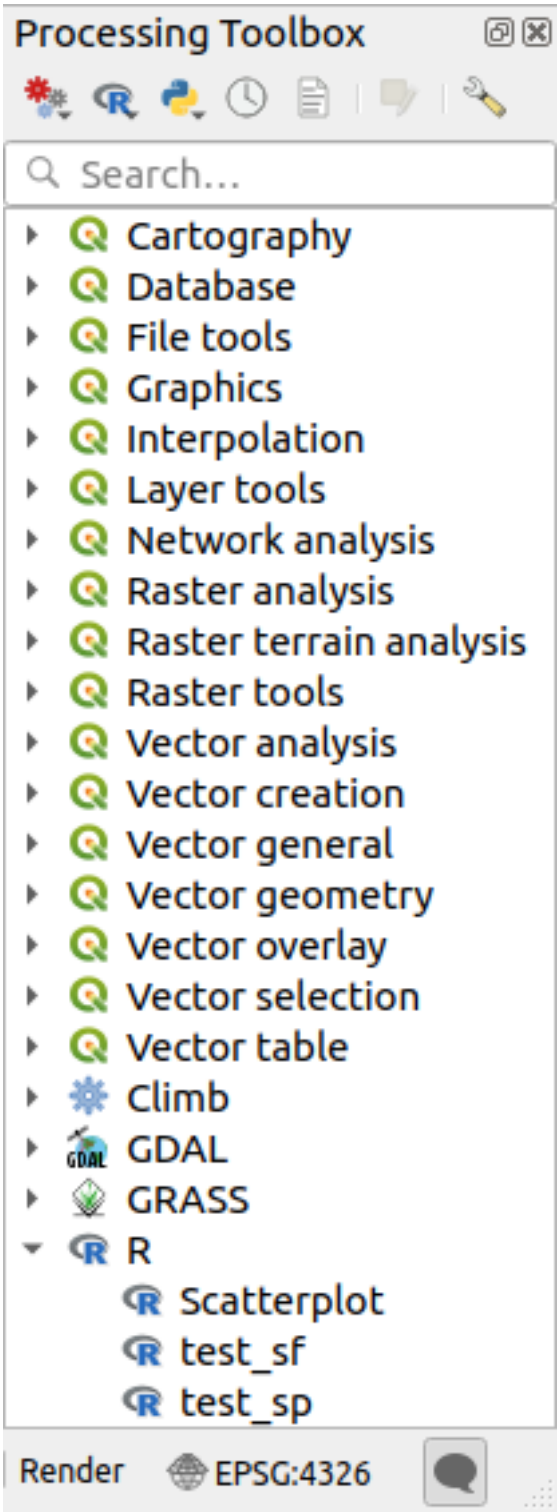
Configuration is done in *Provider* ► *R* in the *Processing* tab of *Settings* ► *Options*.

Depending on your operating system, you may have to use *R folder* to specify where your R binaries are located.

> **Pastaba**
>
> On **Windows** the R executable file is normally in a folder (`R-<version>`) under `C:\Program Files\R\`. Specify the folder and **NOT** the binary!
>
> On **Linux** you just have to make sure that the R folder is in the PATH environment variable. If `R` in a terminal window starts R, then you are ready to go.

After installing the **Processing R Provider** plugin, you will find some example scripts in the *Processing Toolbox*:

- *Scatterplot* runs an R function that produces a scatter plot from two numerical fields of the provided vector layer.
- *test_sf* does some operations that depend on the `sf` package and can be used to check if the R package `sf` is installed. If the package is not installed, R will try to install it (and all the packages it depends on) for you, using the *Package repository* specified in *Provider* ► *R* in the Processing options. The default is https://cran.r-project.org/. Installing may take some time…
- *test_sp* can be used to check if the R package `sp` is installed. If the package is not installed, R will try to install it for you.



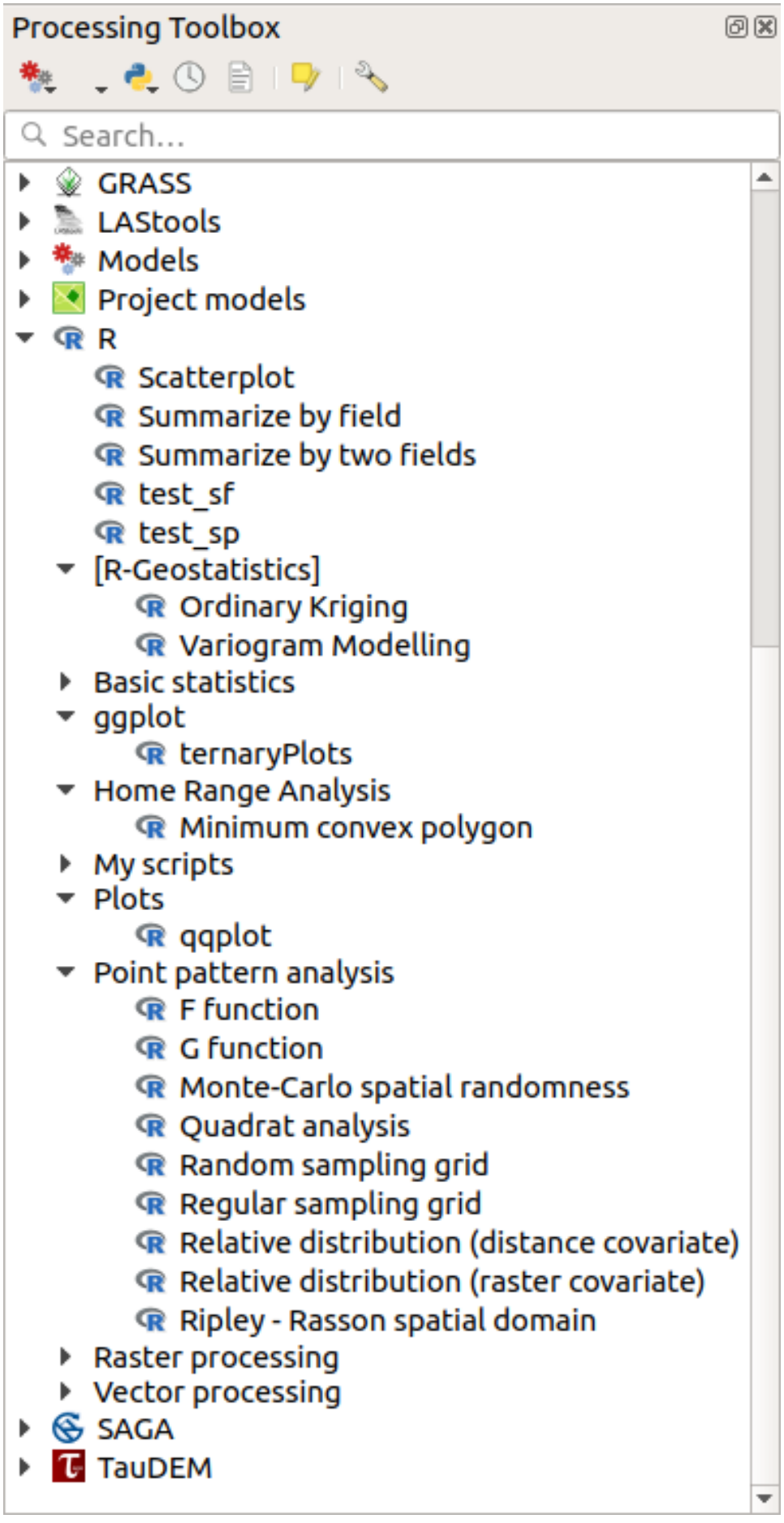
If you have R configured correctly for QGIS, you should be able to run these scripts.

Adding R scripts from the QGIS collection

R integration in QGIS is different from that of some other third party providers like SAGA in that there is not a predefined set of algorithms you can run (except for some example script that come with the *Processing R Provider* plugin).

A set of example R scripts is available in the QGIS Repository. Perform the following steps to load and enable them using the *QGIS Resource Sharing* plugin.

1. Add the *QGIS Resource Sharing* plugin (you may have to enable *Show also experimental plugins* in the Plugin Manager *Settings*)
2. Open it (*Plugins* ► *Resource Sharing* ► *Resource Sharing*)
3. Choose the *Settings* tab
4. Click *Reload repositories*
5. Choose the *All* tab
6. Select *QGIS R script collection* in the list and click on the *Install* button
7. The collection should now be listed in the *Installed* tab
8. Close the plugin
9. Open the *Processing Toolbox*, and if everything is OK, the example scripts will be present under R, in various groups (only some of the groups are expanded in the screenshot below).

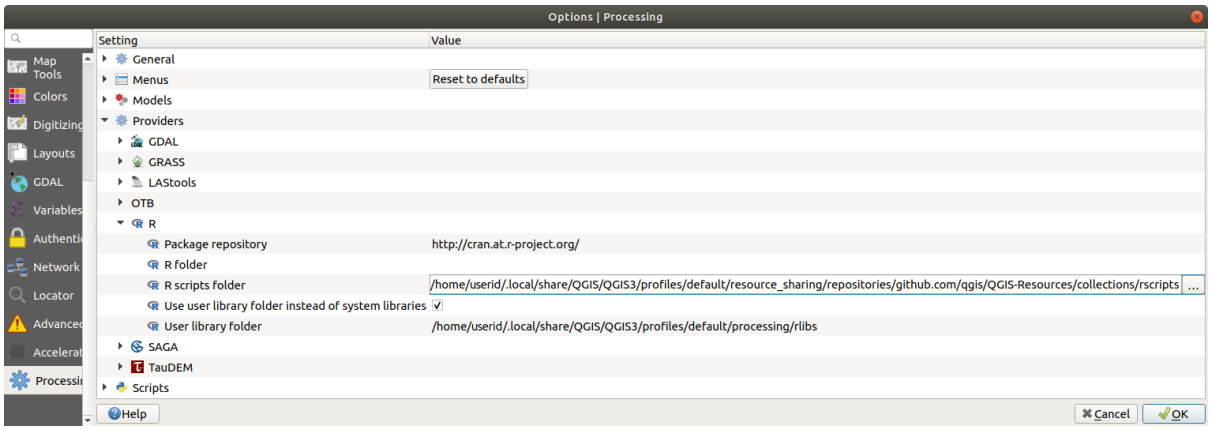


23.35 Fig. : The *Processing Toolbox* with some R scripts shown

The scripts at the top are the example scripts from the *Processing R Provider* plugin.

10. If, for some reason, the scripts are not available in the *Processing Toolbox*, you can try to:
    1. Open the Processing settings (*Settings* ► *Options* ► *Processing* tab)
    2. Go to *Providers* ► *R* ► *R scripts folder*
       - On Ubuntu, set the path to (or, better, include in the path):

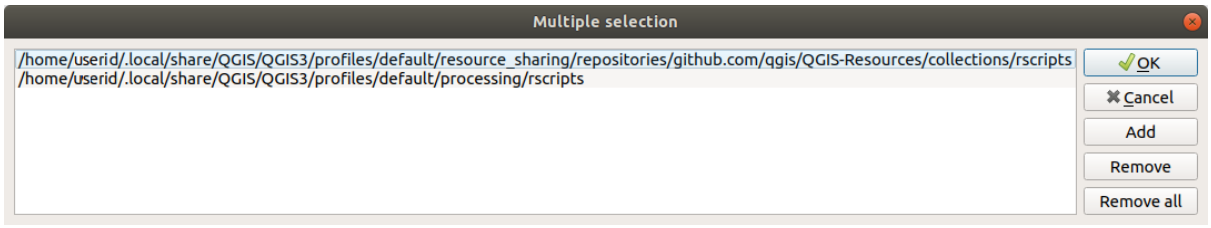
         /home/<user>/.local/share/QGIS/QGIS3/profiles/default/resource_sharing/repositories/github.com/qgis/QGIS-Resources/collections/rscripts



       - On Windows, set the path to (or, better, include in the path):

         C:\Users\<user>\AppData\Roaming\QGIS\QGIS3\profiles\default\resource_sharing\repositories\github.com\qgis\QGIS-Resources\collections\rscripts

    To edit, double-click. You can then choose to just paste / type the path, or you can navigate to the directory by using the … button and press the *Add* button in the dialog that opens. It is possible to provide several directories here. They will be separated by a semicolon („;“).



If you would like to get all the R scrips from the QGIS 2 on-line collection, you can select *QGIS R script collection (from QGIS 2)* instead of *QGIS R script collection*. You will probably find that scripts that depend on vector data input or output will not work.

Creating R scripts

You can write scripts and call R commands, as you would do from R. This section shows you the syntax for using R commands in QGIS, and how to use QGIS objects (layers, tables) in them.

To add an algorithm that calls an R function (or a more complex R script that you have developed and you would like to have available from QGIS), you have to create a script file that performs the R commands.

R script files have the extension `.rsx`, and creating them is pretty easy if you just have a basic knowledge of R syntax and R scripting. They should be stored in the R scripts folder. You can specify the folder (*R scripts folder*) in the *R* settings group in Processing settings dialog).

Let's have a look at a very simple script file, which calls the R method `spsample` to create a random grid within the boundary of the polygons in a given polygon layer. This method belongs to the `maptools` package. Since almost all the algorithms that you might like to incorporate into QGIS will use or generate spatial data, knowledge of spatial packages like `maptools` and `sp`/`sf`, is very useful.

```
##Random points within layer extent=name
##Point pattern analysis=group
##Vector_layer=vector
##Number_of_points=number 10
##Output=output vector
library(sp)
spatpoly = as(Vector_layer, "Spatial")
pts=spsample(spatpoly,Number_of_points,type="random")
spdf=SpatialPointsDataFrame(pts, as.data.frame(pts))
Output=st_as_sf(spdf)
```

The first lines, which start with a double Python comment sign (`##`), define the display name and group of the script, and tell QGIS about its inputs and outputs.

Pastaba

To find out more about how to write your own R scripts, have a look at the R Intro section in the training manual and consult the *QGIS R Syntax* section.

When you declare an input parameter, QGIS uses that information for two things: creating the user interface to ask the user for the value of that parameter, and creating a corresponding R variable that can be used as R function input.

In the above example, we have declared an input of type `vector`, named `Vector_layer`. When executing the algorithm, QGIS will open the layer selected by the user and store it in a variable named `Vector_layer`. So, the name of a parameter is the name of the variable that you use in R for accessing the value of that parameter (you should therefore avoid using reserved R words as parameter names).

Spatial parameters such as vector and raster layers are read using the `st_read()` (or `readOGR`) and `brick()` (or `readGDAL`) commands (you do not have to worry about adding those commands to your description file – QGIS will do it), and they are stored as `sf` (or `Spatial*DataFrame`) objects.

Table fields are stored as strings containing the name of the selected field.

Vector files can be read using the `readOGR()` command instead of `st_read()` by specifying `##load_vector_using_rgdal`. This will produce a `Spatial*DataFrame` object instead of an `sf` object.

Raster files can be read using the `readGDAL()` command instead of `brick()` by specifying `##load_raster_using_rgdal`.

If you are an advanced user and do not want QGIS to create the object for the layer, you can use `##pass_filenames` to indicate that you prefer a string with the filename. In this case, it is up to you to open the file before performing any operation on the data it contains.

With the above information, it is possible to understand the first lines of the R script (the first line not starting with a Python comment character).

```
library(sp)
spatpoly = as(Vector_layer, "Spatial")
pts=spsample(polyg,numpoints,type="random")
```

The `spsample` function is provided by the *sp* library, so the first thing we do is to load that library. The variable `Vector_layer` contains an `sf` object. Since we are going to use a function (`spsample`) from the *sp* library, we must convert the `sf` object to a `SpatialPolygonsDataFrame` object using the `as` function.

Then we call the `spsample` function with this object and the `numpoints` input parameter (which specifies the number of points to generate).

Since we have declared a vector output named `Output`, we have to create a variable named `Output` containing an `sf` object.

We do this in two steps. First we create a `SpatialPolygonsDataFrame` object from the result of the function, using the *SpatialPointsDataFrame* function, and then we convert that object to an `sf` object using the `st_as_sf` function (of the *sf* library).

You can use whatever names you like for your intermediate variables. Just make sure that the variable storing your final result has the defined name (in this case `Output`), and that it contains a suitable value (an `sf` object for vector layer output).

In this case, the result obtained from the `spsample` method had to be converted explicitly into an `sf` object via a `SpatialPointsDataFrame` object, since it is itself an object of class `ppp`, which can not be returned to QGIS.

If your algorithm generates raster layers, the way they are saved will depend on whether or not you have used the `##dontuserasterpackage` option. If you have used it, layers are saved using the `writeGDAL()` method. If not, the `writeRaster()` method from the `raster` package will be used.

If you have used the `##pass_filenames` option, outputs are generated using the `raster` package (with `writeRaster()`).

If your algorithm does not generate a layer, but a text result in the console instead, you have to indicate that you want the console to be shown once the execution is finished. To do so, just start the command lines that produce the results you want to print with the > (‚greater than') sign. Only output from lines prefixed with > are shown. For instance, here is the description file of an algorithm that performs a normality test on a given field (column) of the attributes of a vector layer:

```
##layer=vector
##field=field layer
##nortest=group
library(nortest)
>lillie.test(layer[[field]])
```

The output of the last line is printed, but the output of the first is not (and neither are the outputs from other command lines added automatically by QGIS).

If your algorithm creates any kind of graphics (using the `plot()` method), add the following line (`output_plots_to_html` used to be `showplots`):

```
##output_plots_to_html
```

This will cause QGIS to redirect all R graphical outputs to a temporary file, which will be opened once R execution has finished.

Both graphics and console results will be available through the processing results manager.

For more information, please check the R scripts in the official QGIS collection (you download and install them using the *QGIS Resource Sharing* plugin, as explained in *Adding R scripts from the QGIS collection*). Most of them are rather simple and will greatly help you understand how to create your own scripts.

Pastaba

The `sf`, `rgdal` and `raster` libraries are loaded by default, so you do not have to add the corresponding `library()` commands. However, other libraries that you might need have to be explicitly loaded by typing: `library(ggplot2)` (to load the `ggplot2` library). If the package is not already installed on your machine, Processing will try to download and install it. In this way the package will also become available in R Standalone. **Be aware** that if the package has to be downloaded, the script may take a long time to run the first time.

R libraries installed when running sf_test

The R script `sp_test` tries to load the R packages `sp` and `raster`.

The R script *sf_test* tries to load `sf` and `raster`. If these two packages are not installed, R may try to load and install them (and all the libraries that they depend on).

The following R libraries end up in `~/.local/share/QGIS/QGIS3/profiles/default/processing/rscripts` after `sf_test` has been run from the Processing Toolbox on Ubuntu with version 2.0 of the *Processing R Provider* plugin and a fresh install of *R* 3.4.4 (*apt* package `r-base-core` only):

```
abind, askpass, assertthat, backports, base64enc, BH, bit, bit64, blob, brew,␣
↪callr, classInt, cli, colorspace, covr, crayon, crosstalk, curl, DBI, deldir,
desc, dichromat, digest, dplyr, e1071, ellipsis, evaluate, fansi, farver, fastmap,␣
↪gdtools, ggplot2, glue, goftest, gridExtra, gtable, highr, hms,
htmltools, htmlwidgets, httpuv, httr, jsonlite, knitr, labeling, later, lazyeval,␣
↪leafem, leaflet, leaflet.providers, leafpop, leafsync, lifecycle, lwgeom,
magrittr, maps, mapview, markdown, memoise, microbenchmark, mime, munsell, odbc,␣
↪openssl, pillar, pkgbuild, pkgconfig, pkgload, plogr, plyr, png, polyclip,
praise, prettyunits, processx, promises, ps, purrr, R6, raster, RColorBrewer, Rcpp,
↪ reshape2, rex, rgeos, rlang, rmarkdown, RPostgres, RPostgreSQL,
rprojroot, RSQLite, rstudioapi, satellite, scales, sf, shiny, sourcetools, sp,␣
↪spatstat, spatstat.data, spatstat.utils, stars, stringi, stringr, svglite,
sys, systemfonts, tensor, testthat, tibble, tidyselect, tinytex, units, utf8, uuid,
↪ vctrs, viridis, viridisLite, webshot, withr, xfun, XML, xtable
```

GRASS

Configuring GRASS is very easy. First, the path to the GRASS folder has to be defined, but only if you are running Windows.

By default, the Processing framework tries to configure its GRASS connector to use the GRASS distribution that ships along with QGIS. This should work without problems for most systems, but if you experience problems, you might have to configure the GRASS connector manually. Also, if you want to use a different GRASS installation, you can change the setting to point to the folder where the other version is installed. GRASS 7 is needed for algorithms to work correctly.

If you are running Linux, you just have to make sure that GRASS is correctly installed, and that it can be run without problem from a terminal window.

GRASS algorithms use a region for calculations. This region can be defined manually or automatically, taking the minimum extent that covers all the input layers used to execute the algorithm each time.

LAStools

To use LAStools in QGIS, you need to download and install LAStools on your computer and install the LAStools plugin (available from the official repository) in QGIS.

On Linux platforms, you will need Wine to be able to run some of the tools.

LAStools is activated and configured in the Processing options (*Settings* ► *Options*, *Processing* tab, *Providers* ► *LAStools*), where you can specify the location of LAStools (*LAStools folder*) and Wine (*Wine folder*). On Ubuntu, the default Wine folder is `/usr/bin`.

OTB Applications

OTB applications are fully supported within the QGIS Processing framework.

OTB (Orfeo ToolBox) is an image processing library for remote sensing data. It also provides applications that provide image processing functionalities. The list of applications and their documentation are available in OTB CookBook

Pastaba

Note that OTB is not distributed with QGIS and needs to be installed separately. Binary packages for OTB can be found on the download page.

To configure QGIS processing to find the OTB library:

1. Open the processing settings: *Settings* ► *Options* ► *Processing*
2. You can see `OTB` under *Providers* menu:
    1. Expand the *OTB* entry
    2. Set the *OTB folder*. This is the location of your OTB installation.
    3. Set the *OTB application folder*. This is the location of your OTB applications (`<PATH_TO_OTB_INSTALLATION>/lib/otb/applications`)
    4. Click *OK* to save the settings and close the dialog.

If settings are correct, OTB algorithms will be available in the *Processing Toolbox*.

Documentation of OTB settings available in QGIS Processing

- **OTB folder**: This is the directory where OTB is available.
- **OTB application folder**: This is the location(s) of OTB applications.

  Multiple paths are allowed.
- **Logger level** (optional): Level of logger to use by OTB applications.

  The level of logging controls the amount of detail printed during algorithm execution. Possible values for logger level are `INFO`, `WARNING`, `CRITICAL`, `DEBUG`. This value is `INFO` by default. This is an advanced user configuration.
- **Maximum RAM to use** (optional): by default, OTB applications use all available system RAM.

  You can, however, instruct OTB to use a specific amount of RAM (in MB) using this option. A value of 256 is ignored by the OTB processing provider. This is an advanced user configuration.
- **Geoid file** (optional): Path to the geoid file.

  This option sets the value of the elev.dem.geoid and elev.geoid parameters in OTB applications. Setting this value globally enables users to share it across multiple processing algorithms. Empty by default.
- **SRTM tiles folder** (optional): Directory where SRTM tiles are available.

  SRTM data can be stored locally to avoid downloading of files during processing. This option sets the value of elev.dem.path and elev.dem parameters in OTB applications. Setting this value globally enables users to share it across multiple processing algorithms. Empty by default.

Compatibility and Troubleshoot

Starting from OTB 6.6.1, new releases of OTB are made compatible with at least the latest QGIS version available at that time.

If you have issues with OTB applications in QGIS Processing, please open an issue on the OTB bug tracker, using the `qgis` label.

Additional information about OTB and QGIS can be found in OTB Cookbook.

SKYRIUS 24

Processing providers and algorithms

Processing algorithms and their parameters (as presented in the user interface) are documented here.

24.1 QGIS algorithm provider

QGIS algorithm provider implements various analysis and geoprocessing operations using mostly only QGIS API. So almost all algorithms from this provider will work „out of the box“ without any additional configuration.

This provider incorporates some algorithms from plugins and also adds its own algorithms.

24.1.1 3D kaladėlės

Keičia B3DM į GLTF

Keičia failus iš seno `.B3DM` formato į `.GLTF` ar `.GLB`.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties B3DM	`INPUT`	[3D kaladėlės]	Keičiamas įvesties failas.
Išvesties failas	`OUTPUT`	[3D kaladėlės] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite išvesties 3D kaladėlių failą. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties failas	`OUTPUT`	[3D kaladėlės]	Išvesties sluoksnis `.GLTF` formatu.

Pythono kodas

Algoritmo ID: `native:b3dmtogltf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Keisti GLTF į vektorinius geoobjektus

Keičia `.GLTF` ar `.GLB` failo turinį į standartinius vektorinių sluoksnių formatus.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties GLTF	`INPUT`	[3D kaladėlės]	Keičiamas įvesties failas.
Išvesties poligonai Pasirinktinis	`OUTPUT_POLYGON`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį, į kurį reikia įrašyti poligonų geoobjektus. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Išvesties linijos Pasirinktinis	`OUTPUT_LINES`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį, į kurį reikia įrašyti linijų geoobjektus. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties poligonai	`OUTPUT_POLYGON`	[vektorius: poligonas]	Vektorinis sluoksnis, saugantis įvesties sluoksnio poligono geoobjektus.
Išvesties linijos	`OUTPUT_LINES`	[vektorius: linija]	Vektorinis sluoksnis, saugantis įvesties sluoksnio linijos geoobjektus.

Pythono kodas

Algoritmo ID: `native:gltftovector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.2 Kartografija

Lygiuoti taškus su geoobjektais

Paskaičiuoja pasukimą, kurio reikia, kad taškiniai geoobjektai būtų nukreipti į artimiausią geoobjektą iš kito atskaitos sluoksnio. Į išvesties sluoksnį pridedamas naujas laukas su kampu (laipsniais, pagal laikrodžio rodyklę) iki artimiausio atskaitos geoobjekto.

Pasirinktinai galima keisti išvesties sluoksnio simbologiją, kad ji automatiškai naudotų paskaičiuoto posūkio kampo lauką ir pasuktų žymeklių simbolius. Jei reikia, galima nurodyti maksimalų atstumą, kad būtų išvengta izoliuotų geoobjektų lygiavimo į toli esančius geoobjektus.

Patarimas

Šis algoritmas sukurtas tokiems atvejams kaip pastatų taškų lygiavimas pagal artimiausio kelio kryptį.

Leidžia taškų *geoobjektus keisti vietoje*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: taškai]	Taškų geoobjektai, kurių pasukimą reikia skaičiuoti
Atskaitos sluoksnis	`REFERENCE_LAYE`	[vektorius: geometrija]	Sluoksnis, kuriame reikia ieškoti artimiausio geoobjekto pasukimo skaičiavimui

continues on next page

Table 24.5 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Maksimalus įvertinamas atstumas Pasirinktinis	`MAX_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: jokio	Jei tokiu atstumu nerandamas atskaitos geoobjektas, taško geoobjektui nepriskiriamas joks pasukimas.
Kampo lauko pavadinimas	`FIELD_NAME`	[tekstas] Numatytas: „pasukimas“	Laukas, į kurį reikia įrašyti pasukimo reikšmę.
Automatiškai pritaikyti simbologiją	`APPLY_SYMBOLOG`	[loginis] Numatytas: Taip	Pasuka geoobjektų simbolio žymeklį naudojant kampo lauko reikšmę
Lygiuojamas sluoksnis	`OUTPUT`	[vektorius: taškai] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite pasuktą išvesties vektorinį sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Lygiuojamas sluoksnis	`OUTPUT`	[vektorius: taškai]	Taškų sluoksnis pridedamas su pasukimo lauku. Įkėlus į QGIS, jis taikomas pagal nutylėjimą įvesties sluoksnio simbologijoje naudojant nuo pasukimo duomenų priklausantį žymeklio simbolį.

Pythono kodas

Algoritmo ID: `native:angletonearest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Sujungti stilių duomenų bazes

Sujungia kelias QGIS stilių duomenų bazes į vieną stilių duomenų bazę. Jei skirtingose pradinėse duomenų bazėse yra elementai su tokiu pačiu tipu ir pavadinimu, jei bus pervadinti, kad gautoje sudėtinėje duomenų bazėje turėtų unikalius pavadinimus.

Taip pat žiūrėkite

Create style database from project

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties duomenų bazės	`INPUT`	[file] [list]	Failai su QGIS stilių elementais
Sujungiami objektai	`OBJECTS`	[enumeration] [list]	Stilių elementų tipai įvesties duomenų bazės, kuriuos jūs norite sudėti į naują duomenų bazę. Tai gali būti: • 0 — *Simboliai* • 1 — *Spalvų rampos* • 2 — *Teksto formatai* • 3 — *Užrašų nustatymai*
Išvesties stilių duomenų bazė	`OUTPUT`	[file] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties `.XML` failas su visais parinktais stilių elementais. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Spalvos rampų skaičius	`COLORRAMPS`	[skaičius: sveikas skaičius]	Number of color ramps
Užrašų nustatymų skaičius	`LABELSETTINGS`	[skaičius: sveikas skaičius]	Number of label settings
Išvesties stilių duomenų bazė	`OUTPUT`	[file]	Išvesties `.XML` failas su parinktais stilių elementais
Simbolių skaičius	`SYMBOLS`	[skaičius: sveikas skaičius]	Number of symbols
Teksto formatų skaičius	`TEXTFORMATS`	[skaičius: sveikas skaičius]	Number of text formats

Pythono kodas

Algoritmo ID: `native:combinestyles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Kurti kategorizuotą braižymą pagal stilius

Nustato vektorinio sluoksnio braižymą į kategorizuotą, naudojant atitinkamus simbolius iš stilių duomenų bazės. Jei stiliaus failas nenurodytas, vietoje jo naudojami simboliai iš dabartinio naudotojo *simbolių bibliotekos*.

Nurodyta išraiška ar laukas naudojami kuriant kategirojas braižytojui. Kiekviena kategorija individualiai atitaikoma simboliams, kurie egzistuoja nurodytoje QGIS XML stilių duomenų bazėje. Kai randamas atitinkamo pavadinimo simbolis, kategorijos simbolis nustatomas, kad atitiktų rastą simbolį.

Jei reikia, išvestys taipogi gali būti lentelės su sąrašu kategorijų, kurių nepavyko sutapdinti su simboliais ir simboliai, kurie nebuvo sutapdinti su kategorijomis.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vector: any]	Vektorinis sluoksnis, kuriam reikia taikyti kategorizuotus stilius
Kategorizuoti naudojant išraišką	`FIELD`	[išraiška]	Laukas ar išraiška geoobjektų kategorizavimui
Stilių duomenų bazė (palikite tuščią, jei norite naudoti įrašytus simbolius)	`STYLE`	[file]	Failas (`.XML`) su simboliais, taikomais įvesties sluoksnio kategorijoms. Failą galima gauti naudojant Stilių tvarkyklės įrankiu *Bendrinti simbolius*. Jei failas nenurodytas, naudojami vietiniai QGIS simboliai.
Atsižvelgti į raidžių dydžius atitaikant simbolių pavadinimus	`CASE_SENSITIVE`	[loginis] Numatytas: ne	Jei True (įjungta), atsižvelgia į raidžių dydį, lyginant kategorijų ir simbolių pavadinimus.
Lyginant ignoruoti ne raidžių ir skaitmenų simbolius	`TOLERANT`	[loginis] Numatytas: ne	Jei True (įjungta), ne raidžių ar skaitmenų simboliai kategorijų ir simbolių pavadinimuose bus ignoruojami, taip leidžiant didesnę atitikimo toleranciją.
Neatitinkančios kategorijos Pasirinktinis	`NON_MATCHING_C`	[vektorius: lentelė] Numatytas: `[Praleisti išvestį]`	Išvesties lentelė kategorijoms, kurios neatitinka nei vieno duomenų bazės simbolio. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Neatitinkančių simbolių pavadinimai Pasirinktinis	`NON_MATCHING_S`	[vektorius: lentelė] Numatytas: `[Praleisti išvestį]`	Išvesties lentelė pateiktos simbolių duomenų bazės simboliams, kurie neatitiko nei vienos kategorijos. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Neatitinkančios kategorijos	`NON_MATCHING_C`	[vektorius: lentelė]	Išvardina kategorijas, kurios neatitiko nei vieno pateiktos stilių duomenų bazės simbolio
Neatitinkančių simbolių pavadinimai	`NON_MATCHING_S`	[vektorius: lentelė]	Išvardina pateiktos simbolių duomenų bazės simbolius, kurie neatitiko nei vienos kategorijos
Kategorizuotas sluoksnis	`OUTPUT`	[toks pat, kaip įvestis]	Įvesties vektorinis sluoksnis su pritaikytu kategorizuotu stiliumi. Naujas sluoksnis neišvedamas.

Pythono kodas

Algoritmo ID: `native:categorizeusingstyle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Create style database from project

Extracts all style objects (symbols, color ramps, text formats and label settings) from a QGIS project.

The extracted symbols are saved to a QGIS style database (`XML` format), which can be managed and imported via the *Style Manager* dialog.

Taip pat žiūrėkite

Sujungti stilių duomenų bazes

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input project (leave blank to use current) Pasirinktinis	`INPUT`	[file]	A QGIS project file to extract the style items from
Objects to extract	`OBJECTS`	[enumeration] [list]	Types of style items in the input project you would like to put in the new database. These can be: • 0 — *Simboliai* • 1 — *Spalvų rampos* • 2 — *Teksto formatai* • 3 — *Užrašų nustatymai*

continues on next page

Table 24.11 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties stilių duomenų bazė	`OUTPUT`	[file] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output `.XML` file for the selected style items. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Spalvos rampų skaičius	`COLORRAMPS`	[skaičius: sveikas skaičius]	Number of color ramps
Užrašų nustatymų skaičius	`LABELSETTINGS`	[skaičius: sveikas skaičius]	Number of label settings
Išvesties stilių duomenų bazė	`OUTPUT`	[file]	Output `.XML` file for the selected style items
Simbolių skaičius	`SYMBOLS`	[skaičius: sveikas skaičius]	Number of symbols
Teksto formatų skaičius	`TEXTFORMATS`	[skaičius: sveikas skaičius]	Number of text formats

Pythono kodas

Algorithm ID: `native:stylefromproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export atlas layout as image

Exports the atlas of a print layout as image files (e.g. PNG or JPEG images).

If a coverage layer is set, the selected layout's atlas settings exposed in this algorithm will be overwritten. In this case, an empty filter or sort by expression will turn those settings off.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Atlas layout	`LAYOUT`	[layout]	Layout to export
Coverage layer Pasirinktinis	`COVERAGE_LAYER`	[vektorius: geometrija]	Layer to use to generate the atlas
Filtro išraiška	`FILTER_EXPRESS`	[išraiška]	Expression to use to filter out atlas features

continues on next page

Table 24.13 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Rikiavimo išraiška Pasirinktinis	SORTBY_EXPRESS	[išraiška]	Expression to use to sort the atlas features
Reverse sort order	SORTBY_REVERSE	[loginis] Numatytas: ne	Determines if sorting should be inverted. Used when a sort expression is provided.
Output filename expression	FILENAME_EXPRE	[išraiška] Default: ‚output_'‖@atlas_feat	Expression for use to generate filenames
Output folder	FOLDER	[aplankas]	Destination folder where the images will be generated

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Map layers to assign to unlocked map item(s) Pasirinktinis	LAYERS	[enumeration] [layer]	Layers to display in the map item(s) whose contents are not locked
Image format	EXTENSION	[sąrašas] Default: png	File format of the generated output(s). The list of available formats varies depending on OS and installed drivers.
DPI Pasirinktinis	DPI Numatytas: jokio	[skaičius: slankaus kablelio]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Generate world file	GEOREFERENCE	[loginis] Numatytas: Taip	Determines if a world file should be generated
Export RDF metadata	INCLUDE_METADA	[loginis] Numatytas: Taip	Determines if RDF metadata (title, author, …) should be generated
Enable antialiasing	ANTIALIAS	[loginis] Numatytas: Taip	Determines if antialiasing should be enabled

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Image file	OUTPUT	[file]	Image files generated by the atlas layout

Pythono kodas

Algorithm ID: `native:atlaslayouttoimage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export atlas layout as PDF (multiple files)

Exports the atlas of a print layout to multiple PDF files.

If a coverage layer is set, the selected layout's atlas settings exposed in this algorithm will be overwritten. In this case, an empty filter or sort by expression will turn those settings off.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Atlas layout	`LAYOUT`	[layout]	Layout to export
Coverage layer Pasirinktinis	`COVERAGE_LAYER`	[vektorius: geometrija]	Layer to use to generate the atlas
Filtro išraiška	`FILTER_EXPRESS`	[išraiška]	Expression to use to filter out atlas features
Rikiavimo išraiška Pasirinktinis	`SORTBY_EXPRESS`	[išraiška]	Expression to use to sort the atlas features
Reverse sort order	`SORTBY_REVERSE`	[loginis] Numatytas: ne	Determines if sorting should be inverted. Used when a sort expression is provided.
Output filename Pasirinktinis	`OUTPUT_FILENAM`	[išraiška]	Name pattern of the PDF output files.
Output folder	`OUTPUT_FOLDER`	[aplankas]	Destination folder for the output PDF files.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Map layers to assign to unlocked map item(s) Pasirinktinis	`LAYERS`	[enumeration] [layer]	Layers to display in the map item(s) whose contents are not locked
DPI Pasirinktinis	`DPI` Numatytas: jokio	[skaičius: slankaus kablelio]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Always export as vectors	`FORCE_VECTOR`	[loginis] Numatytas: ne	Determines if vectorial data should be left as vectors
Always export as raster	`FORCE_RASTER`	[loginis] Numatytas: ne	Forces all the items in the map to be rasterized. This parameter takes precedence over the `FORCE_VECTOR` parameter.
Append georeference information	`GEOREFERENCE`	[loginis] Numatytas: Taip	Determines if a world file should be generated
Export RDF metadata	`INCLUDE_METADA`	[loginis] Numatytas: Taip	Determines if RDF metadata (title, author, …) should be generated
Disable tiled raster layer exports	`DISABLE_TILED`	[loginis] Numatytas: ne	Determines if raster should be tiled

continues on next page

Table 24.17 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Simplify geometries to reduce output file size	`SIMPLIFY`	[loginis] Numatytas: Taip	Determines if geometries should be simplified to reduce output file size
Text export	`TEXT_FORMAT`	[sąrašas] Numatytas: 0	Determines if text should be exported as path or text objects. *Possible options* are: • 0 - Always export text as paths (recommended) • 1 - Always export texts as text objects • 2 - Prefer exporting texts as text objects
Image compression	`IMAGE_COMPRESS`	[sąrašas] Numatytas: 0	Determines compression level of the image and how suitable the file could be for printing outputs or post-production in external applications. Possible options are: • 0 - Lossy (JPEG) • 1 - Lossless

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
PDF file	`OUTPUT`	[file]	PDF file corresponding to the exported atlas layout

Pythono kodas

Algorithm ID: `native:atlaslayouttomultiplepdf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export atlas layout as PDF (single file)

Exports the atlas of a print layout as a single PDF file.

If a coverage layer is set, the selected layout's atlas settings exposed in this algorithm will be overwritten. In this case, an empty filter or sort by expression will turn those settings off.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Atlas layout	`LAYOUT`	[layout]	Layout to export
Coverage layer Pasirinktinis	`COVERAGE_LAYER`	[vektorius: geometrija]	Layer to use to generate the atlas
Filtro išraiška	`FILTER_EXPRESS`	[išraiška]	Expression to use to filter out atlas features
Rikiavimo išraiška Pasirinktinis	`SORTBY_EXPRESS`	[išraiška]	Expression to use to sort the atlas features
Reverse sort order	`SORTBY_REVERSE`	[loginis] Numatytas: ne	Determines if sorting should be inverted. Used when a sort expression is provided.
PDF file	`OUTPUT`	[file] Numatytas: [įrašyti į laikiną failą]	Name (including path) of the output file. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Map layers to assign to unlocked map item(s) Pasirinktinis	`LAYERS`	[enumeration] [layer]	Layers to display in the map item(s) whose contents are not locked
DPI Pasirinktinis	`DPI` Numatytas: jokio	[skaičius: slankaus kablelio]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Always export as vectors	`FORCE_VECTOR`	[loginis] Numatytas: ne	Determines if vectorial data should be left as vectors
Always export as raster	`FORCE_RASTER`	[loginis] Numatytas: ne	Forces all the items in the map to be rasterized. This parameter takes precedence over the `FORCE_VECTOR` parameter.
Append georeference information	`GEOREFERENCE`	[loginis] Numatytas: Taip	Determines if a world file should be generated
Export RDF metadata	`INCLUDE_METADA`	[loginis] Numatytas: Taip	Determines if RDF metadata (title, author, …) should be generated
Disable tiled raster layer exports	`DISABLE_TILED`	[loginis] Numatytas: ne	Determines if raster should be tiled
Simplify geometries to reduce output file size	`SIMPLIFY`	[loginis] Numatytas: Taip	Determines if geometries should be simplified to reduce output file size

continues on next page

Table 24.20 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Text export	`TEXT_FORMAT`	[sąrašas] Numatytas: 0	Determines if text should be exported as path or text objects. *Possible options* are: • 0 - Always export text as paths (recommended) • 1 - Always export texts as text objects • 2 - Prefer exporting texts as text objects
Image compression	`IMAGE_COMPRESS`	[sąrašas] Numatytas: 0	Determines compression level of the image and how suitable the file could be for printing outputs or post-production in external applications. Possible options are: • 0 - Lossy (JPEG) • 1 - Lossless

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
PDF file	`OUTPUT`	[file]	PDF file corresponding to the exported atlas layout

Pythono kodas

Algorithm ID: `native:atlaslayouttopdf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export print layout as image

Exports a print layout as an image file (e.g. PNG or JPEG images)

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Print layout	`LAYOUT`	[layout]	Layout to export

continues on next page

Table 24.22 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Image file	`OUTPUT`	[file] Numatytas: [įrašyti į laikiną failą]	Name (including path) of the output file. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Map layers to assign to unlocked map item(s) Pasirinktinis	`LAYERS`	[enumeration] [layer]	Layers to display in the map item(s) whose contents are not locked
DPI Pasirinktinis	`DPI` Numatytas: jokio	[skaičius: slankaus kablelio]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Generate world file	`GEOREFERENCE`	[loginis] Numatytas: Taip	Determines if a world file should be generated
Export RDF metadata	`INCLUDE_METADA`	[loginis] Numatytas: Taip	Determines if RDF metadata (title, author, …) should be generated
Enable antialiasing	`ANTIALIAS`	[loginis] Numatytas: Taip	Determines if antialiasing should be enabled

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Image file	`OUTPUT`	[file]	Image file corresponding to the exported print layout

Pythono kodas

Algorithm ID: `native:printlayouttoimage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export print layout as PDF

Exports a print layout as a PDF file.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Print Layout	`LAYOUT`	[layout]	Layout to export
PDF file	`OUTPUT`	[file] Numatytas: [įrašyti į laikiną failą]	Name (including path) of the output file. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Map layers to assign to unlocked map item(s) Pasirinktinis	`LAYERS`	[enumeration] [layer]	Layers to display in the map item(s) whose contents are not locked
DPI Pasirinktinis	`DPI` Numatytas: jokio	[skaičius: slankaus kablelio]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Always export as vectors	`FORCE_VECTOR`	[loginis] Numatytas: ne	Determines if vectorial data should be left as vectors
Always export as raster	`FORCE_RASTER`	[loginis] Numatytas: ne	Forces all the items in the map to be rasterized. This parameter takes precedence over the `FORCE_VECTOR` parameter.
Append georeference information	`GEOREFERENCE`	[loginis] Numatytas: Taip	Determines if a world file should be generated
Export RDF metadata	`INCLUDE_METADA`	[loginis] Numatytas: Taip	Determines if RDF metadata (title, author, …) should be generated
Disable tiled raster layer exports	`DISABLE_TILED`	[loginis] Numatytas: ne	Determines if raster should be tiled
Simplify geometries to reduce output file size	`SIMPLIFY`	[loginis] Numatytas: Taip	Determines if geometries should be simplified to reduce output file size
Text export	`TEXT_FORMAT`	[sąrašas] Numatytas: 0	Determines if text should be exported as path or text objects. *Possible options* are: • 0 - Always export text as paths (recommended) • 1 - Always export texts as text objects • 2 - Prefer exporting texts as text objects

continues on next page

Table 24.26 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Image compression	`IMAGE_COMPRESS`	[sąrašas] Numatytas: 0	Determines compression level of the image and how suitable the file could be for printing outputs or post-production in external applications. Possible options are: • 0 - Lossy (JPEG) • 1 - Lossless
Export layers as separate PDF files	`SEPARATE_LAYER`	[loginis] Numatytas: ne	If True, then a separate PDF file will be created per layer per map item in the layout. Additionally, separate PDF files may be created for other complex layout items, resulting in a set of PDF files which contain logical atomic components of the layout.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
PDF file	`OUTPUT`	[file]	PDF file(s) corresponding to the exported print layout

Pythono kodas

Algorithm ID: `native:printlayouttopdf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Extract labels

Extracts label information from a rendered map at a given extent and scale.

If a map theme is provided, the rendered map will match the visibility and symbology of that theme. If left blank, all visible layers from the project will be used. Extracted label information include: position (served as point geometries), the associated layer name and feature ID, label text, rotation (in degree, clockwise), multiline alignment, and font details.

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Map extent	`EXTENT`	[apimtis]	Extent of the map to extract the labels from Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Map scale	`SCALE`	[scale]	Extracted labels will be rendered using their properties set at this scale.
Map theme Pasirinktinis	`MAP_THEME`	[maptheme]	A map theme displaying the layers to extract the labels from. If unset, labels of the currently visible layers are extracted.
Include unplaced labels	`INCLUDE_UNPLAC`	[loginis] Numatytas: Taip	Specify whether all overlapping labels should be extracted, including the conflicting (thus unplaced) ones.
Extracted labels	`OUTPUT`	[vektorius: taškai] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer for the extent(s). *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Map resolution (in DPI) Pasirinktinis	`DPI` Default: 96.0	[skaičius: slankaus kablelio]	

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Extracted labels	`OUTPUT`	[vektorius: taškai]	Point vector layer representing the fetched labels. Each feature has attributes identifying its source (layer, feature ID) and the assigned labeling properties (text, font, size, rotation, …). A default style with labeling and null symbol is also applied to the layer. **Įspėjimas** Because some of the generated fields have name with more than 10 characters, using the ESRI shapefile format (`.SHP`) to store the output may lead to unexpected rendering while loading the layer in QGIS.

Pythono kodas

Algorithm ID: `native:extractlabels`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Print layout map extent to layer

Creates a polygon layer containing the extent of a print layout map item (or items), with attributes specifying the map size (in layout units, i.e. the *reference map* units), scale and rotation.

If the map item parameter is specified, then only the matching map extent will be exported. If it is not specified, all map extents from the layout will be exported.

Optionally, a specific output CRS can be specified. If it is not specified, the original map item CRS will be used.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Print layout	`LAYOUT`	[sąrašas]	A print layout in the current project
Map item Pasirinktinis	`MAP`	[sąrašas] Default: *All the map items*	The map item(s) whose information you want to extract. If none is provided then all the map items are processed.

continues on next page

Table 24.31 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Apimtis	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer for the extent(s). *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Overrride CRS Pasirinktinis	`CRS`	[crs] Default: *The layout CRS*	Select the CRS for the layer in which the information will be reported.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Map height	`HEIGHT`	[skaičius: slankaus kablelio]	
Apimtis	`OUTPUT`	[vektorius: poligonas]	Output polygon vector layer containing extents of all the input layout map item(s)
Map rotation	`ROTATION`	[skaičius: slankaus kablelio]	
Map scale	`SCALE`	[skaičius: slankaus kablelio]	
Map width	`WIDTH`	[skaičius: slankaus kablelio]	

Pythono kodas

Algorithm ID: `native:printlayoutmapextenttolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Set layer style

Applies a provided style to a layer. The style must be defined in a `QML` file.

No new output are created: the style is immediately assigned to the layer.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[layer]	Input layer you want to apply the style to
Style file	`STYLE`	[file]	Path to the `.qml` file of the style

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
	`OUTPUT`	[toks pat, kaip įvestis]	The input layer with the new style assigned. No new layer is created.

Pythono kodas

Algorithm ID: `native:setlayerstyle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

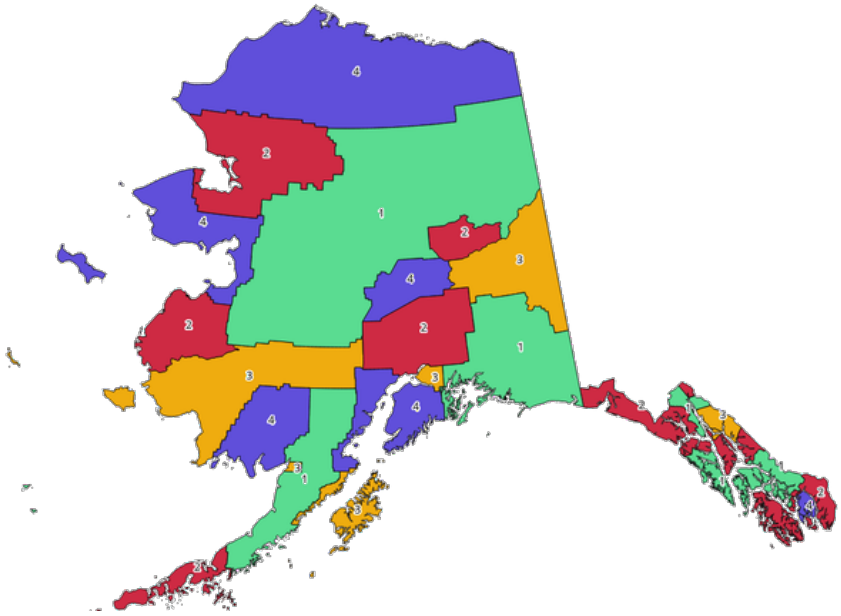
Topological coloring

Assigns a color index to polygon features in such a way that no adjacent polygons share the same color index, whilst minimizing the number of colors required.

The algorithm allows choice of method to use when assigning colors.

A minimum number of colors can be specified if desired. The color index is saved to a new attribute named **color_id**.

The following example shows the algorithm with four different colors chosen; as you can see each color class has the same amount of features.



24.1 Fig. : Topological colors example

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	The input polygon layer
Minimum number of colors	`MIN_COLORS`	[skaičius: sveikas skaičius] Default: 4	The minimum number of colors to assign. Minimum 1, maximum 1000.
Minimum distance between features	`MIN_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Prevent nearby (but non-touching) features from being assigned equal colors. Minimum 0.0.
Balance color assignment	`BALANCE`	[sąrašas] Numatytas: 0	Options are: • 0 — By feature count Attempts to assign colors so that the count of features assigned to each individual color index is balanced. • 1 — By assigned area Assigns colors so that the total area of features assigned to each color is balanced. This mode can be useful to help avoid large features resulting in one of the colors appearing more dominant on a colored map. • 2 — By distance between colors Assigns colors in order to maximize the distance between features of the same color. This mode helps to create a more uniform distribution of colors across a map.

continues on next page

Table 24.36 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Colored	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Colored	`OUTPUT`	[vektorius: poligonas]	Polygon vector layer with an added `color_id` column

Pythono kodas

Algorithm ID: `qgis:topologicalcoloring`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Transfer annotations from main layer

Transfers all *annotations* from the main annotation layer in a project to a new annotation layer. Items placement can then be adjusted within the layer stack.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
New layer name	`LAYER_NAME`	[tekstas] Default: ‚Annotations‘	Name of the annotations layer to create

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
New layer name	`OUTPUT`	[layer]	A layer with items from the main annotation layer

Pythono kodas

Algorithm ID: `native:transferannotationsfrommain`

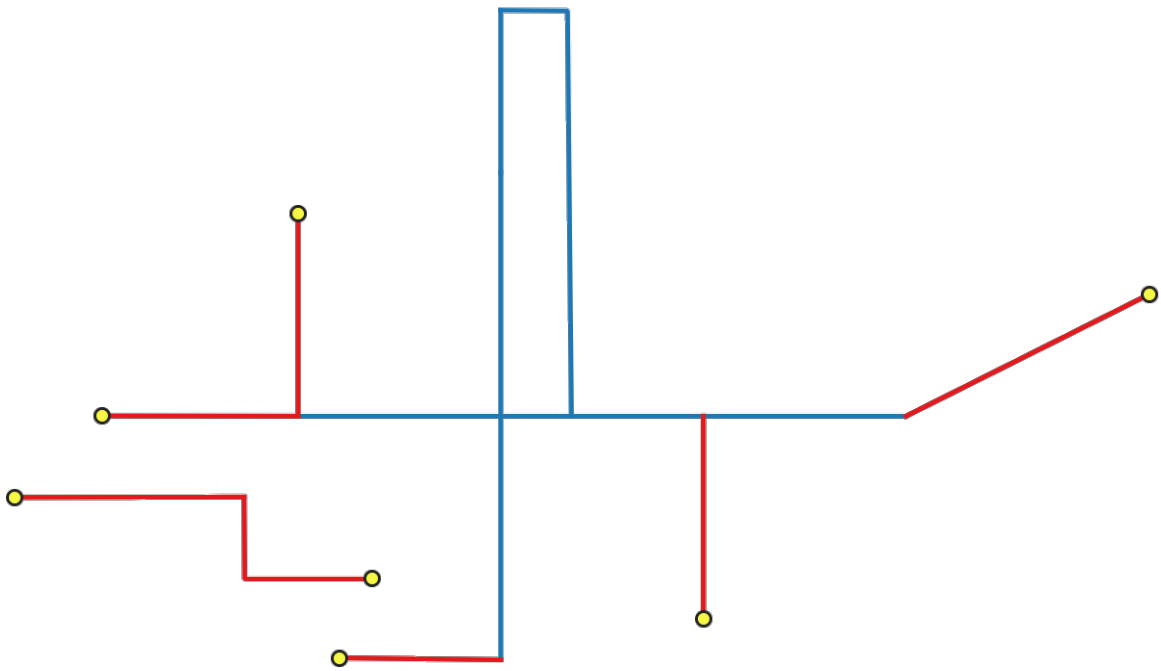
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.3 Check Geometry

Dangle-end lines

Detects dangle-end lines in line geometries and reports them as errors. A dangle-end line is a line feature that terminates at a vertex connected to only one segment, resulting in an endpoint without a proper connection.



24.2 Fig. : Identifying dangle-end features (in red) and vertices (yellow).

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Dangle-end errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Dangle-end features Optional	`OUTPUT`	[vector: line] Default: `[Skip output]`	Line layer containing the identified dangle-end line features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Dangle-end errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the dangle-end. • `gc_ringidx`: the index of the feature's geometry ring containing the dangle-end. • `gc_vertidx`: the vertex index of the dangle-end in the feature's geometry ring. • `gc_errorx`: the x coordinate of the dangle-end. • `gc_errory`: the y coordinate of the dangle-end. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature with the dangle-end.
Dangle-end features	`OUTPUT`	[vector: line]	Output line layer containing the input dangle-end features. If no dangle-end features are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometrydangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Degenerate polygons

Checks the polygons with less than 3 points, which are degenerate polygons. Degenerate polygons are errors.

Pastaba

This algorithm detects invalid polygon geometries (degenerate polygons). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Degenerate polygons errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the centroid of the degenerate polygons. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Degenerate polygons features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Specification of the output layer containing the degenerate polygons. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Degenerate polygons errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the geometry part of the degenerate polygon. • gc_ringidx: the index of the ring of the degenerate polygon. • gc_vertidx • gc_errorx: the x coordinate of the centroid of the degenerate polygon • gc_errory: the y coordinate of the centroid of the degenerate polygon • gc_error • UNIQUE_ID field: the unique ID of the input feature that is degenerate.
Degenerate polygons features	OUTPUT	[vector: polygon]	Output polygon layer with features containing the degenerate polygons. If no degenerate polygon features are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometrydegeneratepolygon

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Duplicated geometries

Checks for duplicated geometries in a vector layer, and reports them as errors.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: geometry]	Layer with the geometries to check
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification
Duplicate geometries errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Duplicate geometries Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer containing the geometries that were found to be duplicated. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Duplicate geometries errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the duplicate geometry. • `gc_errory`: the y coordinate of the centroid of the duplicate geometry. • `gc_error`: the indices of the duplicate geometry (all except the highest index). • `UNIQUE_ID` field: the unique ID of the input feature that is a duplicate.
Duplicate geometries	`OUTPUT`	[vector: same as input]	Output layer containing the geometries that were found to be duplicated. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometryduplicate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Duplicated vertices

Checks for duplicated vertices in line or polygon geometries, and reports them as errors.

Taip pat žiūrėkite

Delete duplicated vertices, *Remove duplicate vertices*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification
Duplicated vertices errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Duplicated vertices features Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer containing the features with duplicated vertices. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Duplicated vertices errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the feature's geometry part containing the duplicate vertex. • gc_ringidx: the index of the feature's geometry ring containing the duplicate vertex. • gc_vertidx: the index of the duplicate vertex in the feature's geometry ring. • gc_errorx: the x coordinate of the duplicate vertex. • gc_errory: the y coordinate of the duplicate vertex. • gc_error • UNIQUE_ID field: the unique ID of the input feature that has duplicate vertices.
Duplicated vertices features	OUTPUT	[vector: same as input]	Output layer containing the features with duplicated vertices. Available fields are the same as in the ERRORS output.

Python code

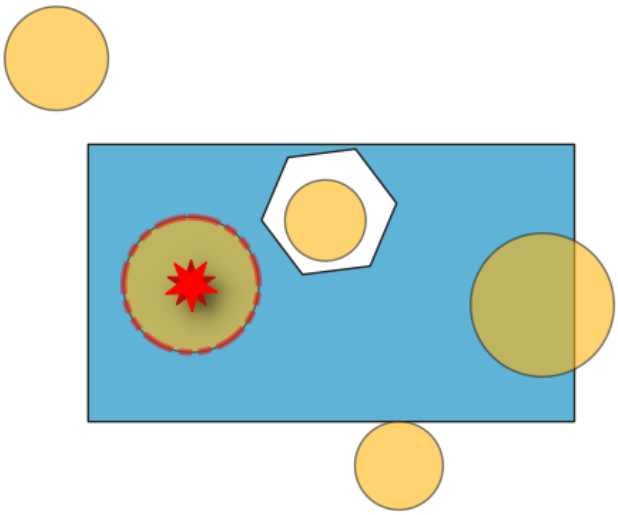
Algorithm ID: native:checkgeometryduplicatenodes

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Features inside polygon

Checks the input geometries contained in the polygons from the polygon layers list. A polygon layer can be checked against itself. Any contained features are reported as errors.



24.3 Fig. : Reporting errors on features inside a polygon.

Taip pat žiūrėkite

Delete features

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: geometry]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Polygon layers	`POLYGONS`	[vector: polygon][list]	List of polygon layers to check against.
Errors from contained features	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Point layer representing the contained features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continues on next page

Table 24.52 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Contained features Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer containing features within the checked polygons. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Errors from contained features	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the contained feature. • `gc_errory`: the y coordinate of the centroid of the contained feature. • `gc_error`: the layer name and feature ID of a polygon that contains the feature. • `UNIQUE_ID` field: the unique ID of the input feature that is contained within another.
Contained features	`OUTPUT`	[vector: same as input]	Output layer containing features within the checked polygons. Available fields are the same as in the `ERRORS` output.

Python code

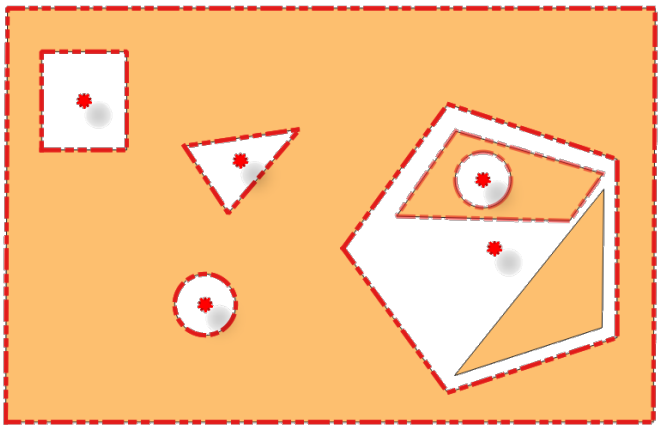
Algorithm ID: `native:checkgeometrycontained`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Holes

Detects holes in polygon geometries and reports them as errors.



24.4 Fig. : Reporting errors for holes on a polygon feature.

Taip pat žiūrėkite

Fill holes

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification

continues on next page

Table 24.55 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Holes errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Polygons with holes Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Polygon layer with the features containing holes. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Holes errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the feature's geometry part containing the hole. • gc_ringidx: the index of the feature's geometry ring containing the hole. • gc_vertidx • gc_errorx: the x coordinate of the centroid of the hole. • gc_errory: the y coordinate of the centroid of the hole. • gc_error • UNIQUE_ID field: the unique ID of the input feature that has a hole.
Polygon with holes	OUTPUT	[vector: polygon]	Output polygon layer with features containing holes. If no holes are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometryhole

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Lines intersecting each other

Checks intersections between line geometries within a layer. Intersections between two different lines are errors.

Taip pat žiūrėkite

Line intersections, *Intersection*, *Self-intersections*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Intersection errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the intersections location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Intersecting feature Optional	`OUTPUT`	[vector: line] Default: `[Skip output]`	Specification of the output layer for features intersecting each other. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Intersection errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part where the intersection occurs. • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the intersection point. • `gc_errory`: the y coordinate of the intersection point. • `UNIQUE_ID` field: the unique ID of an intersecting input feature. • `gc_error`: the layer name and index of the other intersecting feature.
Intersecting feature	`OUTPUT`	[vector: line]	Output layer containing, for each identified intersection, the feature (with same `UNIQUE_ID` value) it belongs to. If no intersections are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometrylineintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Lines intersecting other layer

Checks if the input line layer features intersect with the check layer features. An input feature that intersects with a check layer feature is an error.

Taip pat žiūrėkite

Line intersections, *Intersection*, *Self-intersections*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Check layer	`CHECK_LAYER`	[vector: line, polygon]	Layer with the geometries to check against.
Line intersecting other layer errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output point layer containing the locations of intersection errors. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Line intersecting other layer features Optional	`OUTPUT`	[line] Default: `[Skip output]`	Line layer with the input features that intersect features in the check layer. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Line intersecting other layer errors	ERRORS	[vector: point]	Output point layer representing the error location and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the feature's geometry part where the intersection occurs. • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the intersection point. • gc_errory: the y coordinate of the intersection point. • UNIQUE_ID field: the unique ID of the intersecting input feature. • gc_error: the index of the check layer where the intersection occurs.
Line intersecting other layer features	OUTPUT	[vector: line]	Output layer containing, for each identified intersection, the input feature it belongs to. If no intersections are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

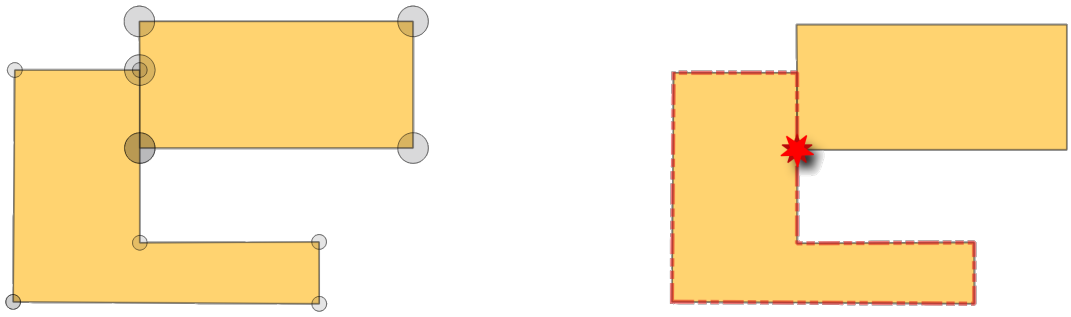
Algorithm ID: native:checkgeometrylinelayerintersection

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Missing vertices along borders

Checks for missing vertices along polygon borders. To be topologically correct, a vertex at the junction of two polygons must be present on both polygons. Missing vertices are errors.



24.5 Fig. : Reporting errors for missing vertices on polygon features.

Taip pat žiūrėkite

Add missing vertices along borders

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Missing vertices errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Missing vertices features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Polygon layer with the features whose vertices are missing. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	TOLERANCE	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Missing vertices errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the missing vertex. • gc_errory: the y coordinate of the missing vertex. • gc_error • UNIQUE_ID field: the unique ID of the input feature that has a missing vertex.
Missing vertices features	OUTPUT	[vector: polygon]	Output polygon layer with features whose vertices are missing. There will be as many (duplicate) features as there are missing vertices in each geometry. If no vertices are missing, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

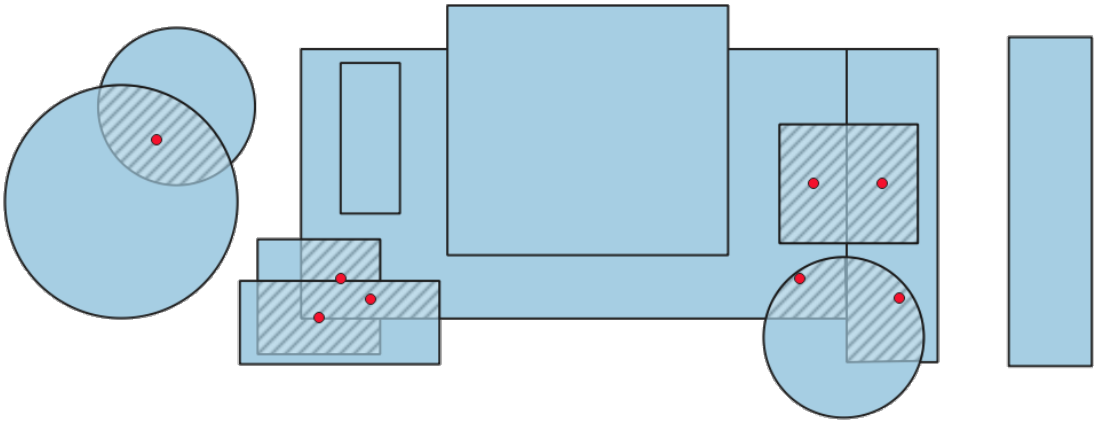
Algorithm ID: native:checkgeometrymissingvertex

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Overlaps

Calculates overlapping areas in polygon geometries, and reports areas smaller than a given minimum as errors.



24.6 Fig. : Hashed polygons indicate overlapping areas smaller than the specified minimum.

Taip pat žiūrėkite

Delete overlaps, *Validate coverage*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Minimum overlap area	`MIN_OVERLAP_AR`	[numeric: double]	Minimum area of the overlap to be reported as an error. If set to 0, all overlaps are reported.
Overlap errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing centroid points of the overlapping areas. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continues on next page

Table 24.67 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Overlap features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Specification of the output layer for features containing overlaps. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Overlap errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_errorx`: the x coordinate of the centroid of the overlapping area. • `gc_errory`: the y coordinate of the centroid of the overlapping area. • `gc_error`: the area of the overlapping geometry. • `UNIQUE_ID` field: the unique ID of an overlapped input feature. • `gc_overlap_feature_{unique_id}`: the `UNIQUE_ID` field value for the other overlapping feature.
Overlap features	`OUTPUT`	[vector: polygon]	Output layer containing the overlapping areas. Available fields are the same as in the `ERRORS` output.

Python code

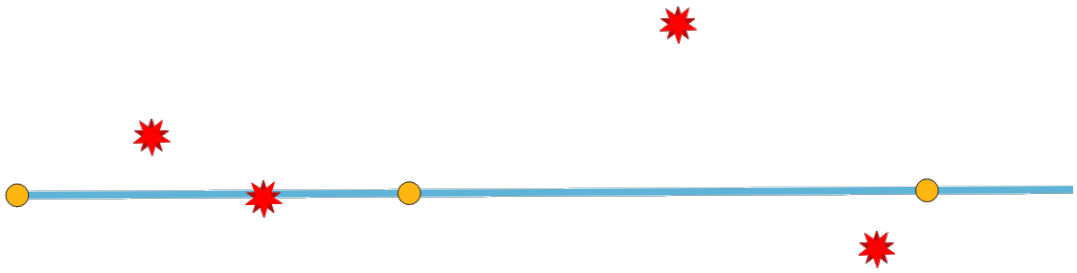
Algorithm ID: `native:checkgeometryoverlap`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Points outside lines

Checks if the points in the input layer are covered by a line in the selected line layers. A point not covered by a line is an error.



24.7 Fig. : Reporting errors on points not covered by a line.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: point]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Line layers	`LINES`	[vector: line][list]	Layer(s) with the lines to check against.
Points not covered by a line	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing points not covered by a line. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Points not covered by a line	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of point outside the lines. • `gc_errory`: the y coordinate of point outside the lines. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature that is outside the lines.

Python code

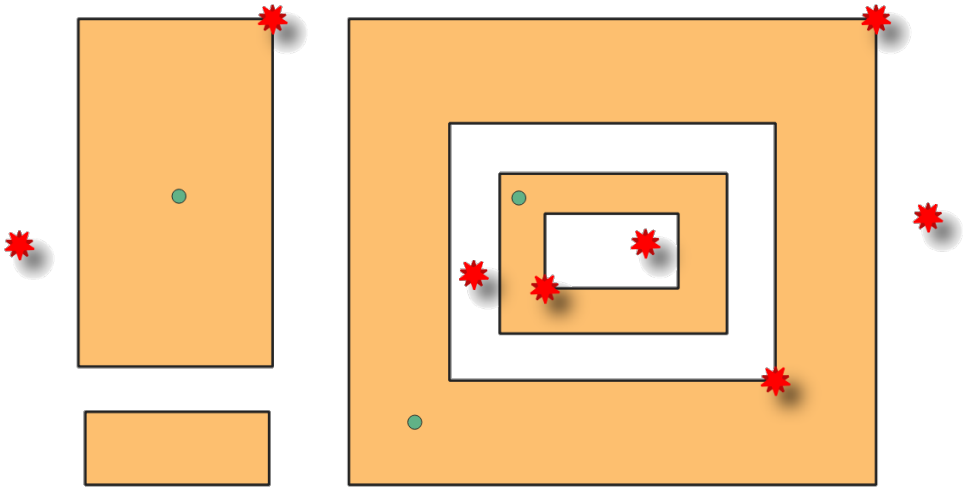
Algorithm ID: `native:checkgeometrypointcoveredbyline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Points outside polygons

Checks if points from the input layer are in polygons from the selected polygon layers. Points that are not fully inside polygons are errors.



24.8 Fig. : Reporting errors on points outside polygons.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: point]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Polygon layers	`POLYGONS`	[vector: polygon][list]	Layer(s) with the polygons to check against.
Points outside polygons errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing points outside the polygons. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Points outside polygons errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of point outside the polygons. • `gc_errory`: the y coordinate of point outside the polygons. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature that is outside the polygons.

Python code

Algorithm ID: `native:checkgeometrypointinpolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

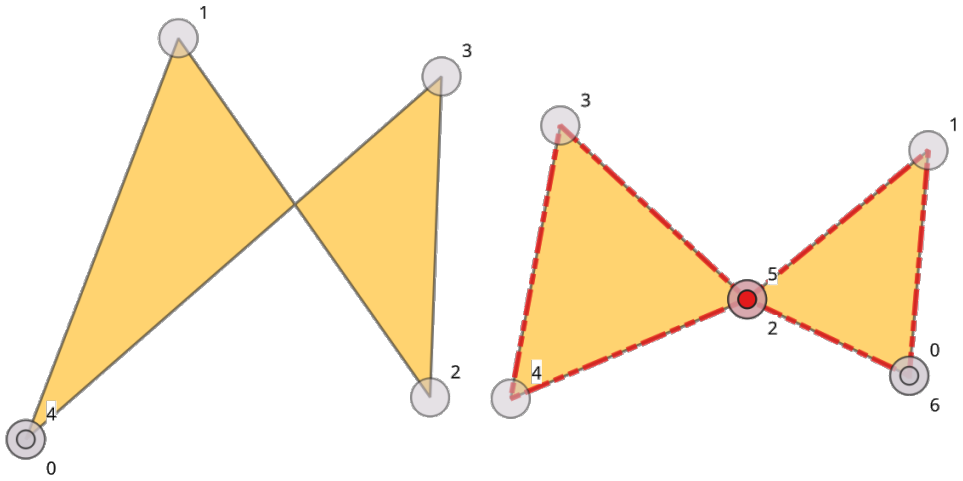
The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Self-contacts

Checks if the geometry has self contact points (in line or polygon), i.e., a vertex that touches more than two segments of the same ring. Self contacts are errors.

Pastaba

This algorithm detects invalid polygon geometries (self-contacts). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.



24.9 Fig. : Self-intersection vs self-contact.

Taip pat žiūrėkite

Self-intersections

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.

continues on next page

Table 24.76 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Self contact error points	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Self contact features Optional	`OUTPUT`	[same as input] Default: `[Skip output]`	Line or polygon layer containing self-contact features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Self contact error points	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the feature's geometry part where the self-contact occurs. • gc_ringidx: the index of the ring where the self-contact occurs. • gc_vertidx: the index of the vertex where the self-contact occurs. • gc_errorx: the x coordinate of the self-contact. • gc_errory: the y coordinate of the self-contact. • gc_error • UNIQUE_ID field: the unique ID of the self-contacting input feature.
Self contact features	OUTPUT	[vector: same as input]	Output polygon or line layer with features containing the self-contact features. If no self-contact features are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometryselfcontact

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

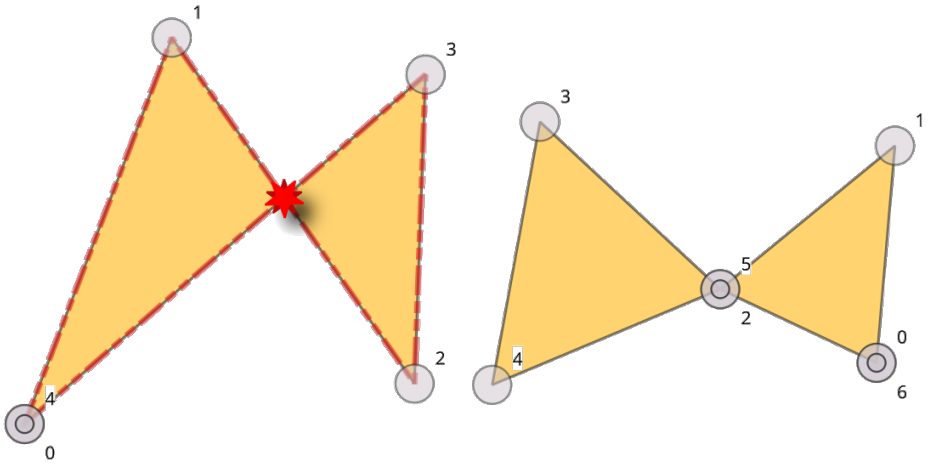
The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Self-intersections

Detects self-intersections in line or polygon geometries, and reports them as errors. Self-intersections occur when the segments of a geometry cross over each other without having a common vertex.

Pastaba

This algorithm detects invalid polygon geometries (self-intersections). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to Do not Filter (Better Performance).



24.10 Fig. : Self-intersection vs self-contact.

Taip pat žiūrėkite

Split self-intersecting geometries, *Self-contacts*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification
Self-intersecting errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Self-intersecting features Optional	`OUTPUT`	[same as input] Default: `[Skip output]`	Line or polygon layer containing self-intersected features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Self-intersecting errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part where the self-intersection occurs. • `gc_ringidx`: the index of the feature's geometry ring where the self-intersection occurs. • `gc_vertidx` • `gc_errorx`: the x coordinate of the self-intersection. • `gc_errory`: the y coordinate of the self-intersection. • `gc_error` • `gc_segment_1`: the index of the first segment involved in the intersection. • `gc_segment_2`: the index of the second segment involved in the intersection. • `UNIQUE_ID` field: the unique ID of the self-intersecting input feature.
Self-intersecting features	`OUTPUT`	[vector: line, polygon]	Output polygon or line layer with features containing the self-intersecting features. If no self-intersecting features are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometryselfintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

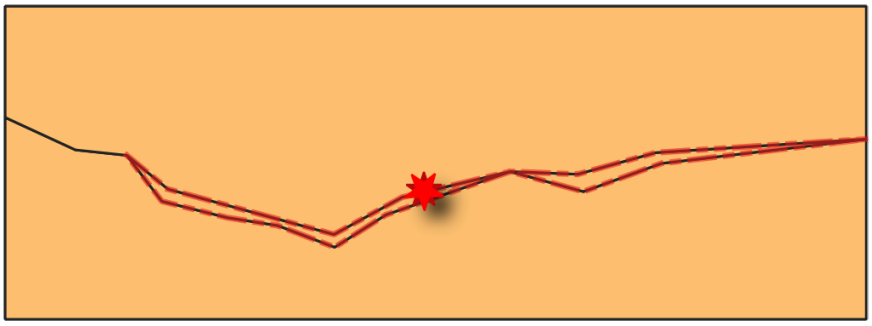
Sliver polygons

Detects sliver polygons in a polygon layer, i.e., polygons with a thinness greater than a specified value.

The thinness is the ratio between the area of the minimum square containing the polygon and the area of the polygon itself (a square has a thinness value of 1). The thinness value is between 1 and +infinity. A maximum area can be set for limiting the checks to polygons of a lower area. Polygons having a thinness higher than the maximum thinness are errors. To fix sliver polygons, use the *Fix small polygons* algorithm.

Pastaba

This algorithm detects invalid polygon geometries (sliver polygons). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.



24.11 Fig. : Reporting errors on features with sliver polygons.

Taip pat žiūrėkite

Fix small polygons

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Maximum thinness	`MAX_THINNESS`	[numeric: double] Default: 20.0	Maximum thinness value of a polygon to be considered valid. Any polygon with a thinness value above this threshold is reported as error.
Maximum area (map units squared)	`MAX_AREA`	[numeric: double] Default: 0.0	Maximum area of a polygon to be checked (in map units). Any polygon with an area above this threshold is skipped. A value of 0 means all polygons are checked.
Sliver polygon errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the centroid points of sliver polygon features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Sliver polygon features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Polygon layer containing sliver polygon features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Sliver polygon errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the centroid of the sliver polygon feature. • gc_errory: the y coordinate of the centroid of the sliver polygon feature. • gc_error: the thinness value of the sliver polygon feature. • UNIQUE_ID field: the unique ID of the input feature that is a sliver polygon.
Sliver polygon features	OUTPUT	[vector: polygon]	Output layer with features containing the sliver polygons. If no sliver polygon features are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

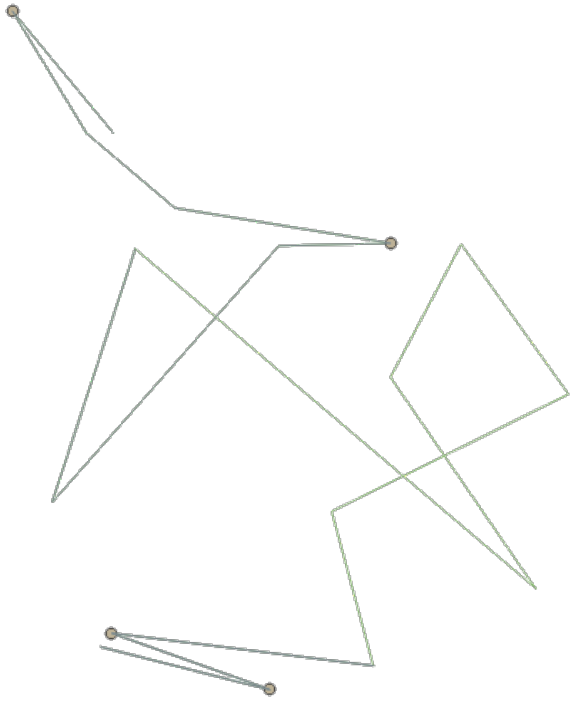
Algorithm ID: native:checkgeometrysliverpolygon

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Small angles

Compares the angles within line or polygon geometries to a specified threshold, and reports as error any angle below that value.



24.12 Fig. : Reporting errors on a line feature for angles lower than 15°.

Taip pat žiūrėkite

Delete small angles

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Minimum angle	`MIN_ANGLE`	[numeric: double]	Allowed minimum angle between adjacent segments in a feature's geometry. Any angle below this threshold is reported as error.

continues on next page

Table 24.85 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Small angle errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Small angle errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the small angle. • `gc_ringidx`: the index of the feature's geometry ring containing the small angle. • `gc_vertidx`: the index of the vertex with the small angle in the feature's geometry ring. • `gc_errorx`: the x coordinate of the vertex with the small angle. • `gc_errory`: the y coordinate of the vertex with the small angle. • `gc_error`: the error angle value. • `UNIQUE_ID` field: the unique ID of the input feature with the small angle.

Python code

Algorithm ID: `native:checkgeometryangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

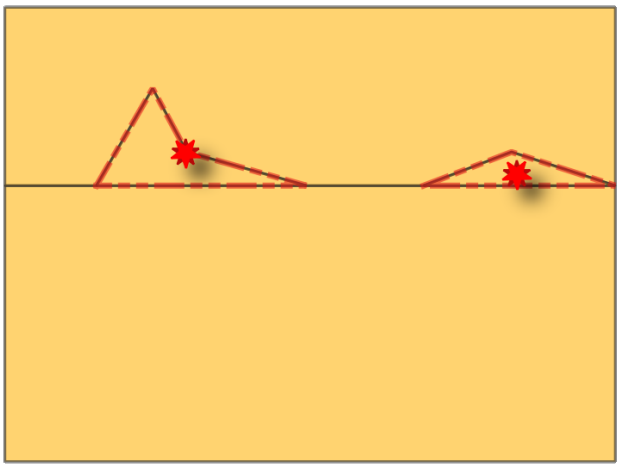
The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Small polygons

Detects polygon features whose area is below a specified value as errors.

Pastaba

This algorithm detects invalid polygon geometries (small polygons). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.



24.13 Fig. : Reporting errors on features with small polygons (below given area threshold).

Taip pat žiūrėkite

Fix small polygons

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Area threshold	`AREATHRESHOLD`	[numeric: double] Default: 0	Minimum area of a polygon to be considered valid, in selected units. Any polygon with an area below this threshold is reported as error.
Small polygons errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing centroid points of small polygons. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Small polygons features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Specification of the output layer for features containing small polygon features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Small polygons errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the centroid of the small polygon. • gc_errory: the y coordinate of the centroid of the small polygon. • gc_error: the area of the small polygon. • UNIQUE_ID field: the unique ID of the input feature that is a small polygon.
Small polygons features	OUTPUT	[vector: polygon]	Output polygon layer with features containing the small polygon features. If no small polygon features are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

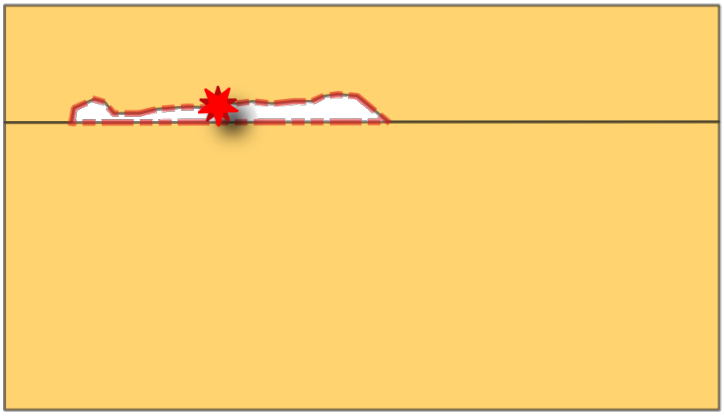
Algorithm ID: native:checkgeometryarea

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Small gaps

Checks for gaps between polygons in the input layer. Gaps with an area smaller than the gap threshold are reported as errors. If an allowed gaps layer is provided, the algorithm ignores gaps that fall entirely within polygons from this layer. An optional buffer can be applied to the allowed gaps.



24.14 Fig. : Reporting errors on polygon features for gaps smaller than the specified threshold.

Taip pat žiūrėkite

Fill gaps, *Validate coverage*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Gap threshold	`GAP_THRESHOLD`	[numeric: double] Default: 0.0	Maximum area of gaps to be reported as errors, in map units. If set to 0, all the gaps are reported.
Allowed gaps layer Optional	`ALLOWED_GAPS_L`	[vector: polygon]	Optional layer specifying polygons whose areas should be ignored during the gap check. Gaps that fall entirely within these polygons will not be reported as errors.
Allowed gaps buffer Optional	`ALLOWED_GAPS_B`	[numeric: double] Default: 0.0	Buffer distance to apply to the allowed gaps layer, in selected units. Gaps located within this buffered area are ignored. Allows adding a spatial tolerance around allowed gaps to avoid reporting near-boundary gaps as errors.

continues on next page

Table 24.91 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Neighbors layer	`NEIGHBORS`	[vector: table] Default: `[Create temporary layer]`	Specification of the output table representing the gap ID and the unique ID of its neighbor features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Gap errors Optional	`ERRORS`	[vector: point] Default: `[Skip output]`	Specification of the output layer containing the centroid points of the gaps. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Gap features	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing the gap geometries. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Neighbors layer	`NEIGHBORS`	[vector: table]	A 1–N relational table, meaning that one gap can be associated with multiple neighboring polygons. The output table contains the following fields: • `gc_errorid`: the ID of the gap. • `UNIQUE_ID` field: the unique ID of an input feature that is a neighbor of the gap.

continues on next page

Table 24.93 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Gap errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the gap. • `gc_errory`: the y coordinate of the centroid of the gap. • `gc_error`: the area of the gap. • `gc_errorid`: the ID of the gap.
Gap features	`OUTPUT`	[vector: polygon]	Output layer containing the gap geometries. Available fields are the same as in the `ERRORS` output.

Python code

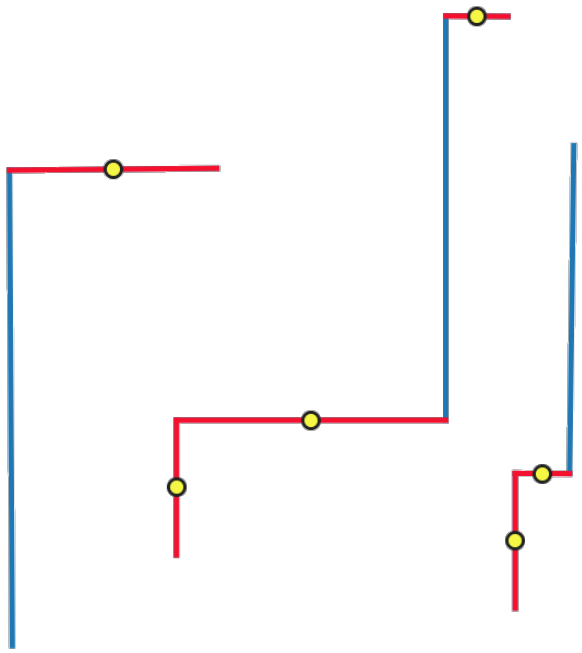
Algorithm ID: `native:checkgeometrygap`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Small segments

Calculates length of individual segments in line or polygon geometries, and reports segments shorter than a minimum length as errors.



24.15 Fig. : Reporting errors (in red) on single line segments shorter than the given threshold.

Taip pat žiūrėkite

Delete features

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Minimum segment length	`MIN_SEGMENT_LE`	[numeric: double] Default: 0.0	Minimum length of segments to be considered valid, in map units.
Short segments errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing centroid of short segments. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continues on next page

Table 24.94 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Short segments features Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer for features containing short segments. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Short segments errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the small segment. • `gc_ringidx`: the index of the feature's geometry ring containing the small segment. • `gc_vertidx`: the index of the end vertex of the small segment in the feature's geometry ring. • `gc_errorx`: the x coordinate of the centroid of the small segment. • `gc_errory`: the y coordinate of the centroid of the small segment. • `gc_error`: the error segment length. • `UNIQUE_ID` field: the unique ID of the input feature with the small segment.

continues on next page

Table 24.96 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Short segments features	`OUTPUT`	[vector: same as input]	Output layer containing, for each identified short segment, the feature it belongs to. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometrysegmentlength`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Strictly multipart

Checks if multipart geometries have more than one part. Multipart geometries with only one part are errors.

Taip pat žiūrėkite

Convert to strictly multipart

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: geometry]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
One-part geometry errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing centroid of the multipart geometries that have only one part. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continues on next page

Table 24.97 – tęsinys iš praeito puslapio

Label	Name	Type	Description
One-part geometry features Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer for features containing multipart geometries that have only one part. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
One-part geometry errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the multipart geometry that has only one part. • `gc_errory`: the y coordinate of the centroid of the multipart geometry that has only one part. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature that is multipart but has only one part.
One-part geometry features	`OUTPUT`	[vector: same as input]	Output layer containing features that are multipart but have only one part. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometrymultipart`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

24.1.4 Duomenų bazės

Eksportuoti į PostgreSQL

Eksportuoja vektorinį sluoksnį į PostgreSQL duomenų bazę, sukuriant naują lenttelę. Jei lentelė tokiu pavadinimu jau yra, ją galima išimti prie kuriant naują. Prieš vykdant šį veiksmą turi būti sukurta jungtis tarp QGIS ir PostgreSQL duomenų bazės (žr. *Įrašytos jungties kūrimas*).

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Importuojamas sluoksnis	`INPUT`	[vektorius: bet koks]	Į duomenų bazę pridedamas vektorinis sluoksnis
Duomenų bazė (jungties pavadinimas)	`DATABASE`	[tekstas]	Duomenų bazės jungties pavadinimas (ne duomenų bazės pavadinimas). Iškrentančiame sąraše bus rodomos esamos jungtys.
Schema (schemos pavadinimas) Pasirinktinis	`SCHEMA`	[tekstas] Numatyta: ‚public‘	Pavadinimas schemos, kurioje reikia laikyti duomenis. Tai gali būti nauja arba jau esama schema.
Lentelė, į kurią importuojama (palikite tuščią, jei norite naudoti sluoksnio pavadinimą) Pasirinktinis	`TABLENAME`	[tekstas] Numatytas: ‚‘	Nustato importuojamo vektorinio failo lentelės pavadinimą. Jei niekas nenurodyta, bus naudojamas sluoksnio pavadinimas.
Pirminio rakto laukas Pasirinktinis	`PRIMARY_KEY`	[tablefield: any]	Nustato pirminio rakto lauką pagal esamą vektorinio sluoksnio lauką. Stulpelis su **unikaliomis** reikšmėmis gali būti naudojamas kaip duomenų bazės Pirminis raktas.
Geometrijos stulpelis	`GEOMETRY_COLUM`	[tekstas] Numatytas: ‚geom‘	Defines the name of the geometry column in the new PostgreSQL table. Geometry information for the features is stored in this column.
Koduotė Pasirinktinis	`ENCODING`	[tekstas] Numatyta: ‚UTF-8‘	Nustato išvesties sluoksnio koduotę

continues on next page

Table 24.100 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Perrašyti	`OVERWRITE`	[boolean] Numatytas: Taip	Jei nurodyta lentelė jau yra, nustačius šios parinkties reikšmę `True` užtikrins, kad ji bus ištrinta ir nauja lentelė sukurta prieš pridedant geoobjektus. Jei šios parinkties reikšmė `False` ir lentelė yra, algoritmas praneš klaidą („lentelė jau yra“).
Kurti erdvinį indeksą	`CREATEINDEX`	[boolean] Numatytas: Taip	Nurodo kurti erdvinį indeksą ar ne
Keisti laukų pavadinimus į mažąsias raides	`LOWERCASE_NAME`	[boolean] Numatytas: Taip	Keičia įvesties vektorinio sluoksnio laukų pavadinimus į mažąsias raides
Išmesti tekstinių laukų ilgio apribojimus	`DROP_STRING_LE`	[boolean] Numatytas: ne	Ar reikia išmesti tekstinių laukų ilgio apribojimus
Kurti vienos dalies geometrijas vietoje kelių dalių	`FORCE_SINGLEPA`	[boolean] Numatytas: ne	Ar išvesties sluoksnio geoobjektai turėtų būti vienos dalies vietoj kelių dalių. Pagal nutylėjimą išlaikoma esama geometrijų informacija.

Išvestys

Algoritmas neturi išvesties.

Pythono kodas

Algorithm ID: `native:importintopostgis`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export to SpatiaLite

Exports a vector layer to a SpatiaLite database. Prior to this a connection between QGIS and the SpatiaLite database has to be created (see eg *SpatiaLite sluoksniai*).

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Importuojamas sluoksnis	`INPUT`	[vektorius: bet koks]	Į duomenų bazę pridedamas vektorinis sluoksnis
File database	`DATABASE`	[vektorius: bet koks]	The SQLite/SpatiaLite database file to connect to

continues on next page

Table 24.101 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Lentelė, į kurią importuojama (palikite tuščią, jei norite naudoti sluoksnio pavadinimą) Pasirinktinis	TABLENAME	[tekstas] Numatytas: ‚'	Defines the table name for the imported vector file. If nothing is specified, the layer name will be used.
Pirminio rakto laukas Pasirinktinis	PRIMARY_KEY	[tablefield: any]	Use a field in the input vector layer as the primary key
Geometrijos stulpelis	GEOMETRY_COLUM	[tekstas] Numatytas: ‚geom'	Defines the name of the geometry column in the new SpatiaLite table. Geometry information for the features is stored in this column.
Koduotė Pasirinktinis	ENCODING	[tekstas] Numatyta: ‚UTF-8'	Nustato išvesties sluoksnio koduotę
Perrašyti	OVERWRITE	[boolean] Numatytas: Taip	If the specified table exists, setting this option to `True` will make sure that it is deleted and a new table will be created before the features of the layer is added. If this option is `False` and the table exists, the algorithm will throw an exception („table already exists").
Kurti erdvinį indeksą	CREATEINDEX	[boolean] Numatytas: Taip	Nurodo kurti erdvinį indeksą ar ne
Keisti laukų pavadinimus į mažąsias raides	LOWERCASE_NAME	[boolean] Numatytas: Taip	Convert the field names of the input vector layer to lowercase
Išmesti tekstinių laukų ilgio apribojimus	DROP_STRING_LE	[boolean] Numatytas: ne	Ar reikia išmesti tekstinių laukų ilgio apribojimus
Kurti vienos dalies geometrijas vietoje kelių dalių	FORCE_SINGLEPA	[boolean] Numatytas: ne	Ar išvesties sluoksnio geoobjektai turėtų būti vienos dalies vietoj kelių dalių. Pagal nutylėjimą išlaikoma esama geometrijų informacija.

Išvestys

Algoritmas neturi išvesties.

Pythono kodas

Algorithm ID: `native:importintospatialite`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Package layers

Adds layers to a GeoPackage.

If the GeoPackage exists and `Overwrite existing GeoPackage` is checked, it will be overwritten (removed and recreated). If the GeoPackage exists and `Overwrite existing GeoPackage` is not checked, the layer will be appended.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`LAYERS`	[vektorius: bet koks] [sąrašas]	The (vector) layers to import into the GeoPackage. Raster layers are not supported. If a raster layer is added, a `QgsProcessingException` will be thrown.
Overwrite existing GeoPackage	`OVERWRITE`	[boolean] Numatytas: ne	If the specified GeoPackage exists, setting this option to `True` will make sure that it is deleted and a new one will be created before the layers are added. If set to `False`, layers will be appended.
Save layer styles into GeoPackage	`SAVE_STYLES`	[boolean] Numatytas: Taip	Save the layer styles.
Save only selected features	`SELECTED_FEATU`	[boolean] Numatytas: ne	If a layer has a selection, setting this option to `True` will result in only selected features being saved. For layers without a selection all features will be saved.
Export related layers following relations defined in the project	`EXPORT_RELATED`	[boolean] Numatytas: ne	If an input layer has *relations* set in the project, setting this option to `True` will result in exporting also its related layer(s). If the layer has features selected, then only their related features will be exported unless the related layer was also an input layer.
Destination GeoPackage	`OUTPUT`	[file] Numatytas: `[Įrašyti į laikiną failą]`	Specify where to store the GeoPackage file. One of • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Layers within new package	`OUTPUT_LAYERS`	[string] [list]	The list of layers added to the GeoPackage.

Pythono kodas

Algorithm ID: `native:package`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

PostgreSQL execute and load SQL

Allows a SQL database query to be performed on a PostgreSQL database connected to QGIS and loads the result. The algorithm **won't** create a new layer: it is designed to run queries on the layer itself.

Pavyzdys

1. Set all the values of an existing field to a fixed value. The SQL query string will be:

   ```
   UPDATE your_table SET field_to_update=20;
   ```

   In the example above, the values of the field `field_to_update` of the table `your_table` will be all set to `20`.

2. Create a new `area` column and calculate the area of each feature with the `ST_AREA` PostGIS function.

   ```
   -- Create the new column "area" on the table your_table"
   ALTER TABLE your_table ADD COLUMN area double precision;
   -- Update the "area" column and calculate the area of each feature:
   UPDATE your_table SET area=ST_AREA(geom);
   ```

Taip pat žiūrėkite

PostgreSQL execute SQL, *Vykdyti SQL*, *SpatiaLite execute SQL*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Duomenų bazė (jungties pavadinimas)	`DATABASE`	[tekstas]	The database connection (not the database name). Existing connections will be shown in the combobox.
SQL query	`SQL`	[tekstas]	Defines the SQL query, for example `'UPDATE my_table SET field=10'`.
Unique ID field name	`ID_FIELD`	[tekstas] Default: id	Sets the primary key field (a column in the result table)
Geometry field name Pasirinktinis	`GEOMETRY_FIELD`	[tekstas] Numatytas: ‚geom‘	Name of the geometry column (a column in the result table)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
SQL layer	`OUTPUT`	[vektorius: bet koks]	The resulting vector layer to be loaded into QGIS.

Pythono kodas

Algorithm ID: `qgis:postgisexecuteandloadsql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

PostgreSQL execute SQL

Allows a SQL database query to be performed on a PostgreSQL database connected to QGIS. The algorithm **won't** create a new layer: it is designed to run queries on the layer itself.

Pavyzdys

1. Set all the values of an existing field to a fixed value. The SQL query string will be:

   ```
   UPDATE your_table SET field_to_update=20;
   ```

   In the example above, the values of the field `field_to_update` of the table `your_table` will be all set to `20`.

2. Create a new `area` column and calculate the area of each feature with the `ST_AREA` PostGIS function.

   ```
   -- Create the new column "area" on the table your_table"
   ALTER TABLE your_table ADD COLUMN area double precision;
   -- Update the "area" column and calculate the area of each feature:
   UPDATE your_table SET area=ST_AREA(geom);
   ```

Taip pat žiūrėkite

PostgreSQL execute and load SQL, *Vykdyti SQL*, *SpatiaLite execute SQL*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Duomenų bazė (jungties pavadinimas)	`DATABASE`	[tekstas]	The database connection (not the database name). Existing connections will be shown in the combobox.
SQL query	`SQL`	[tekstas]	Defines the SQL query, for example `'UPDATE my_table SET field=10'`.

Išvestys

No output is created. The SQL query is executed in place.

Pythono kodas

Algorithm ID: `native:postgisexecutesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

SpatiaLite execute SQL

Allows a SQL database query to be performed on a SpatiaLite database. The database is determined by an input layer or file. The algorithm **won't** create a new layer: it is designed to run queries on the layer itself.

Taip pat žiūrėkite

PostgreSQL execute SQL, *Vykdyti SQL*

For some SQL query examples see *PostgreSQL Query Examples*.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Database layer (or file)	`DATABASE`	[file]	The SQLite/SpatiaLite database file to connect to
SQL query	`SQL`	[tekstas] Numatytas: ‚‘	Defines the SQL query, for example `'UPDATE my_table SET field=10'`.

Išvestys

No output is created. The SQL query is executed in place.

Pythono kodas

Algorithm ID: `native:spatialiteexecutesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

SpatiaLite execute SQL (registered DB)

Allows a SQL database query to be performed on a SpatiaLite database connected to QGIS. The algorithm **won't** create a new layer: it is designed to run queries on the layer itself.

Taip pat žiūrėkite

PostgreSQL execute SQL, *Vykdyti SQL*

For some SQL query examples see *PostgreSQL Query Examples*.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Database	`DATABASE`	[sąrašas] Default: not set	Select a SQLite/SpatiaLite database connected to the current session
SQL query	`SQL`	[tekstas] Numatytas: ‚'	Defines the SQL query, for example `'UPDATE my_table SET field=10'`.

Išvestys

No output is created. The SQL query is executed in place.

Pythono kodas

Algorithm ID: `native:spatialiteexecutesqlregistered`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.5 Failų įrankiai

Atsiųsti failą per HTTP(S)

Atsiunčia URL į failų sistemą naudojant HTTP(S) GET ar POST užklausą.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
URL	`URL`	[tekstas]	Failo, kurį reikia atsiųsti, URL.
Failo paskirtis	`OUTPUT`	[tekstas] Numatytas: `[Įrašyti į laikiną failą]`	Paskirties failo specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Metodas	`METHOD`	[sąrašas] Numatytas: 0	Užklausai naudojamas HTTP metodas. Parinktys yra: • 0 — GET • 1 — POST
Duomenys Pasirinktinis	`DATA`	[tekstas]	Duomenys, kuriuos reikia pridėti, jei užklausa yra POST.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Failo paskirtis	`OUTPUT`	[tekstas]	Atsiųsto failo vieta

Pythono kodas

Algorithm ID: `native:filedownloader`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

HTTP(S) POST/GET request

Performs a HTTP(S) POST/GET request and returns the HTTP status code and the reply data. If an error occurs then the error code and the message will be returned. Optionally, the result can be written to a file on the disk. By default the algorithm will warn on errors. Optionally, the algorithm can be set to treat HTTP errors as failures.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
URL or file path	`URL`	[tekstas]	The URL or path of the file to open.
Metodas	`METHOD`	[sąrašas] Numatytas: 0	Užklausai naudojamas HTTP metodas. Parinktys yra: • 0 — GET • 1 — POST
POST Data Pasirinktinis	`DATA`	[tekstas]	Duomenys, kuriuos reikia pridėti, jei užklausa yra POST.
Authentication Pasirinktinis	`AUTH_CONFIG`	[authconfig] Default: No authentication	An authentication configuration to pass
Consider HTTP errors as failures	`FAIL_ON_ERROR`	[boolean] Numatytas: ne	If set, the algorithm will fail on encountering an HTTP error.
Failo paskirtis Pasirinktinis	`OUTPUT`	[tekstas] Default: `[Skip Output]`	The result can be written to a file instead of being returned as a string. Specification of the file destination. *One of*: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Failo paskirtis	`OUTPUT`	[file]	The downloaded file with the returned result
HTTP Status	`STATUS_CODE`	[tekstas]	The HTTP Status
Network error code	`ERROR_CODE`	[tekstas]	The error code when it fails (like as well on 404 HTTP Status etc.)
Network error message	`ERROR_MESSAGE`	[tekstas]	A string containing the error message in case of failure
Result data	`RESULT_DATA`	[tekstas]	A string containing the result data in case of success

Pythono kodas

Algorithm ID: `native:httprequest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Open file or URL

Opens files in their default associated application, or URLs in the user's default web browser.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
URL or file path	`URL`	[tekstas]	The URL or path of the file to open.

Išvestys

Algoritmas neturi išvesties.

Pythono kodas

Algorithm ID: `native:openurl`

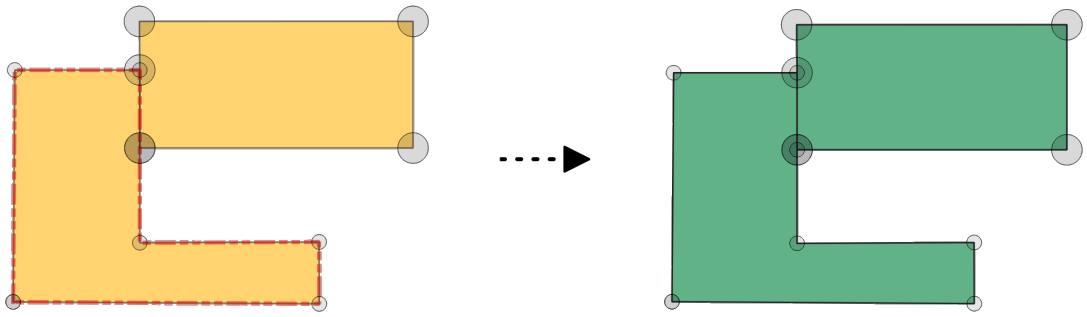
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.6 Fix Geometry

Add missing vertices along borders

Adds the missing vertices along polygons borders, based on an error layer from the *Missing vertices along borders* algorithm.



24.16 Fig. : Fixing missing vertices along polygon borders.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Border vertices fixed layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continues on next page

Table 24.115 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Report layer from fixing border vertices	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing border vertices	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the missing vertex. • `gc_errory`: the y coordinate of the missing vertex. • `gc_error` • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed. • `UNIQUE_ID` field: the unique ID of the input feature that has a missing vertex.
Border vertices fixed layer	`OUTPUT`	[vector: polygon]	Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometrymissingvertex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Convert to strictly multipart

Converts multipart geometries that consists of only one geometry into singlepart geometries, based on an error layer from the *Strictly multipart* algorithm. Layer geometry type will not be changed and will remain multipart.

Taip pat žiūrėkite

Promote to multipart

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Strictly-multipart layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing multipart	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the centroids of the fixed geometries. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing multipart	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the multipart geometry that has only one part. • `gc_errory`: the y coordinate of the centroid of the multipart geometry that has only one part. • `gc_error` • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed. • `UNIQUE_ID` field: the unique ID of the input feature that is multipart but has only one part.
Strictly-multipart layer	`OUTPUT`	[vector: same as input]	Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometrymultipart`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Delete duplicated vertices

Deletes duplicated vertices from the input geometries, based on errors reported by the *Duplicated vertices* algorithm.

Taip pat žiūrėkite

Remove duplicate vertices

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Fixed duplicate vertices layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing duplicate vertices	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing duplicate vertices	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the duplicate vertex. • `gc_ringidx`: the index of the feature's geometry ring containing the duplicate vertex. • `gc_vertidx`: the index of the duplicate vertex in the feature's geometry ring. • `gc_errorx`: the x coordinate of the duplicate vertex. • `gc_errory`: the y coordinate of the duplicate vertex. • `gc_error` • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed. • `UNIQUE_ID` field: the unique ID of the input feature that has duplicate vertices.
Fixed duplicate vertices layer	`OUTPUT`	[vector: same as input]	Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryduplicatenodes`

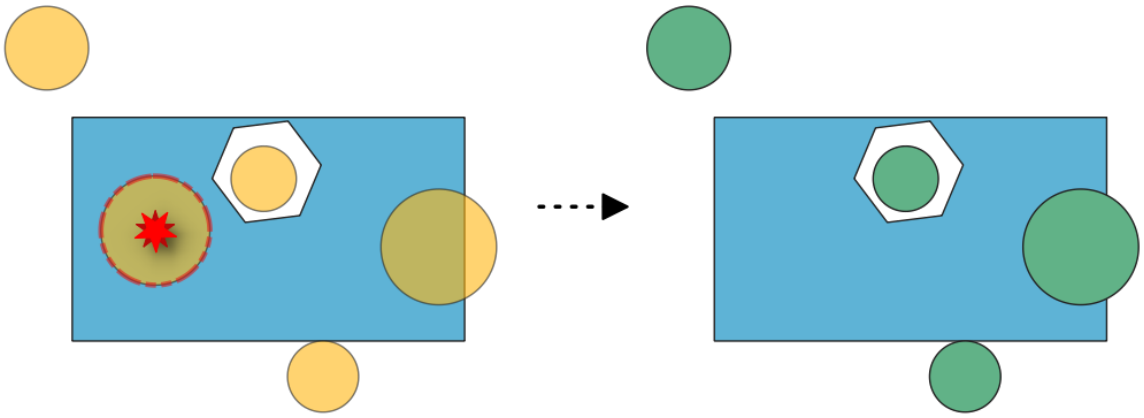
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Delete features

Deletes error features based on an error layer from some check geometry algorithms, such as:

- *Features inside polygon*
- *Degenerate polygons*
- *Small segments*
- *Duplicated geometries*



24.17 Fig. : Fixing errors from the „*Features inside polygon*“ algorithm check.

Taip pat žiūrėkite

Delete duplicate geometries, *Drop geometries*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: geometry]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.

continues on next page

Table 24.124 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Cleaned layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from deleting features	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Outputs

Label	Name	Type	Description
Report layer from deleting features	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the erroneous geometry. • `gc_errory`: the y coordinate of the centroid of the erroneous geometry. • `gc_error`: the index of the feature where the error belongs. • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed. • `UNIQUE_ID` field: the unique ID of the input feature with the error.

continues on next page

Table 24.125 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Cleaned layer	`OUTPUT`	[vector: same as input]	Output layer with features removed based on detected errors. The layer contains the same fields as in the input layer.

Python code

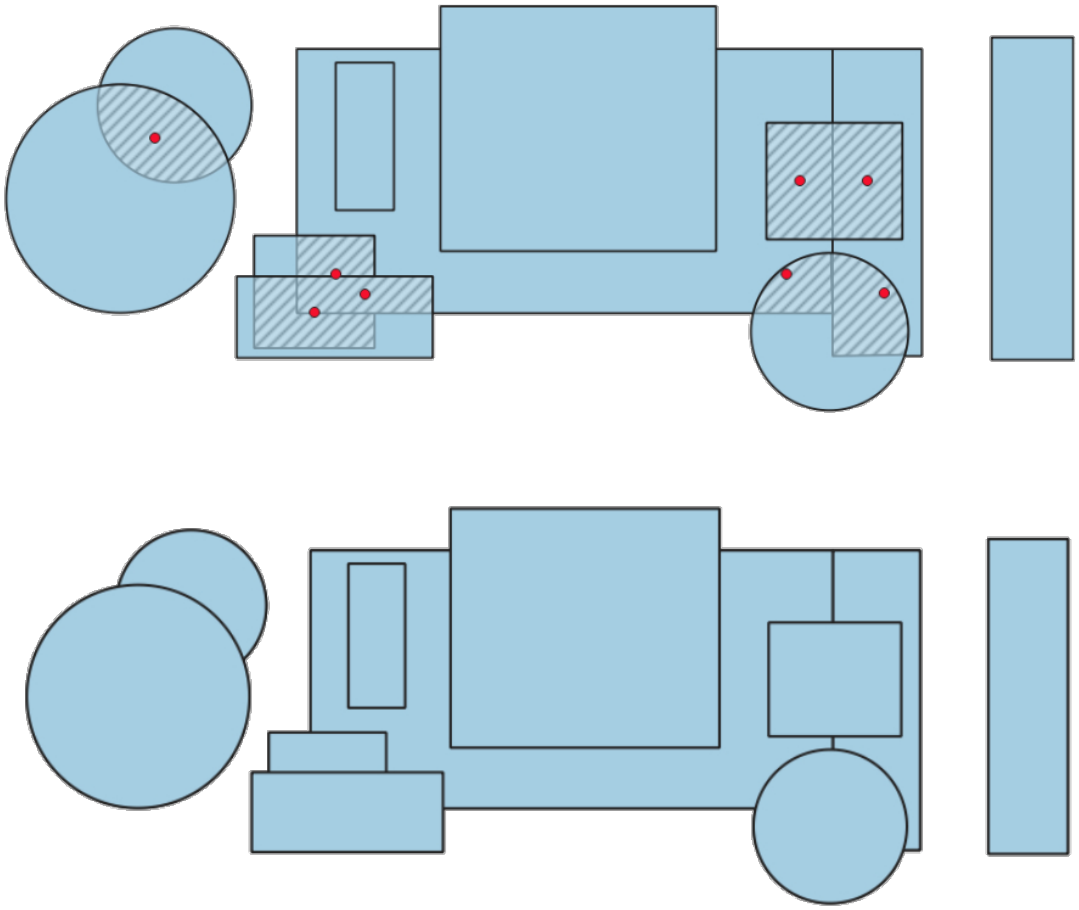
Algorithm ID: `native:fixgeometrydeletefeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Delete overlaps

Deletes overlapping areas based on an error layer from the *Overlaps* algorithm.



24.18 Fig. : Deleting overlapping areas.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of overlap feature unique identifier	`OVERLAP_FEATUR`	[tablefield: any]	Field storing unique values for the overlapping feature identification in input layer, as set in the check algorithm.
Field of error value	`ERROR_VALUE_ID`	[tablefield: any]	Field storing the error value for each feature, as set in the check algorithm.
No-overlap layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing overlaps	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing overlaps	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_errorx`: the x coordinate of the centroid of the overlapping area. • `gc_errory`: the y coordinate of the centroid of the overlapping area. • `gc_error`: the area of the overlapping geometry. • `UNIQUE_ID` field: the unique ID of the unique ID of an overlapped input feature. • `gc_overlap_feature_{unique_id}`: the `UNIQUE_ID` field value for the other overlapping feature. • `report`: a text field describing the fix applied or justifying the failure. Possible values are: – Remove overlapping area from neighboring polygon with shortest shared edge – Error is obsolete • `error_fixed`: a boolean field indicating whether the error was fixed.
No-overlap layer	`OUTPUT`	[vector: polygon]	Output layer with input features edited. Overlapping areas reported as errors are removed. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryoverlap`

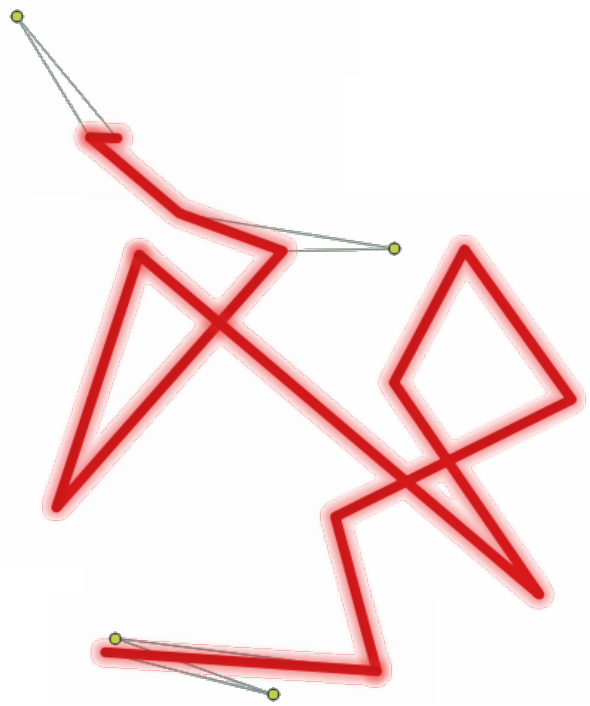
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Delete small angles

Deletes vertices based on an error layer from the *Small angles* algorithm. When deletion of a vertex results in a duplicate vertex (when a spike vertex is deleted), the duplicate vertex is deleted to keep a single vertex and preserve topology.

> **Dėmesio**
>
> This algorithm removes the vertex at the reported small angles, generating new segments that may form a new small angle.



24.19 Fig. : Fixing reported errors on a line feature for angles lower than 15°.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.

continues on next page

Table 24.129 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Small angle fixed layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing small angles	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing small angles	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the small angle. • `gc_ringidx`: the index of the feature's geometry ring containing the small angle. • `gc_vertidx`: the index of the vertex with the small angle in the feature's geometry ring. • `gc_errorx`: the x coordinate of the vertex with the small angle. • `gc_errory`: the y coordinate of the vertex with the small angle. • `gc_error`: the error angle value. • `UNIQUE_ID` field: the unique ID of the input feature with the small angle. • `report`: a text field describing the fix applied or justifying the failure. • `error_fixed`: a boolean field indicating whether the error was fixed.
Small angle fixed layer	`OUTPUT`	[same as input]	Output layer with the geometry fix applied to the input features The layer contains the same fields as in the input layer.

Python code

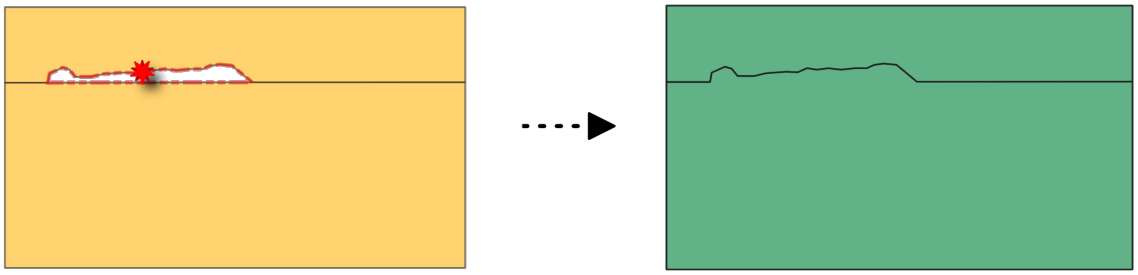
Algorithm ID: `native:fixgeometryangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Fill gaps

Fills the gaps based on a gap and neighbors layer from the *Small gaps* algorithm. Three different fixing methods are available.



24.20 Fig. : Fixing gap between polygon features using the `Add to largest neighbouring area` method.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Neighbors layer	`NEIGHBORS`	[vector: table]	Relational table with the unique ID of the gap and its neighbors in input layer, generated by the check algorithm.
Gaps layer	`GAPS`	[vector: polygon]	Polygon layer with the gaps geometries, generated by the check algorithm.
Method	`METHOD`	[enumeration] Default: 0	Method to apply to the gap fixing. Possible values are: • 0 — `Add to longest shared edge`: adds the gap to the neighboring feature with the longest shared edge. • 1 — `Create new feature`: creates a new feature for each gap. • 2 — `Add to largest neighbouring area`: adds the gap to the neighboring feature with the largest area.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any] Default: gc_errorid	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of error id	`ERROR_ID_IDX`	[tablefield: any]	Field storing the error ID for each gap, as output by the check algorithm.

continues on next page

Table 24.132 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Gaps-filled layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing gaps	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing gaps	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the gap. • `gc_errory`: the y coordinate of the centroid of the gap. • `gc_error`: the area of the gap. • `gc_errorid`: the ID of the gap. • `report`: a text field mentioning the method used to fix the gap. • `error_fixed`: a boolean field indicating whether the error was fixed.
Gaps-filled layer	`OUTPUT`	[vector: polygon]	Output layer of polygons without gaps. The layer contains the same fields as in the input layer.

Python code

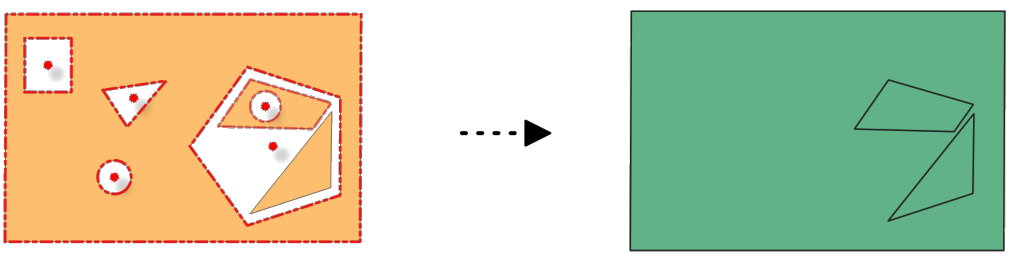
Algorithm ID: `native:fixgeometrygap`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Fill holes

Deletes holes in polygon geometries based on an error layer from the *Holes* algorithm.



24.21 Fig. : Deleting holes in a polygon feature.

Taip pat žiūrėkite

Delete holes

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Holes-filled layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing holes	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing holes	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the hole. • `gc_ringidx`: the index of the feature's geometry ring containing the hole. • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the hole. • `gc_errory`: the y coordinate of the centroid of the hole. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature that has a hole. • `report`: a text field describing the fix applied or justifying the failure. • `error_fixed`: a boolean field indicating whether the error was fixed.
Holes-filled layer	`OUTPUT`	[vector: polygon]	Output layer of polygons without holes. The layer contains the same fields as in the input layer.

Python code

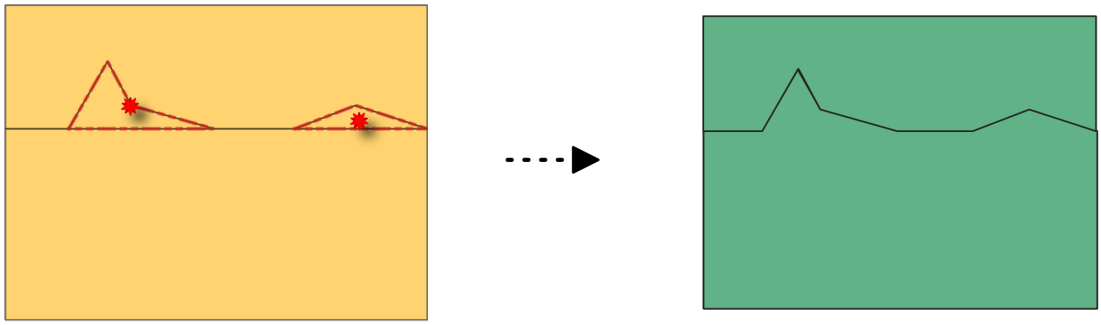
Algorithm ID: `native:fixgeometryhole`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Fix small polygons

Merges neighboring polygons according to the chosen method, based on an error layer from the *Small polygons* or *Sliver polygons* algorithm.



24.22 Fig. : Before and after fixing small polygons (method: merge with neighboring polygon with largest area).

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Method	`METHOD`	[enumeration] Default: 0	Method to apply to the small polygon geometries: • 0 — `Merge with neighboring polygon with longest shared edge`: merges the small polygon geometry with the neighboring polygon that has the longest shared edge. • 1 — `Merge with neighboring polygon with largest area`: merges the small polygon geometry with the neighboring polygon that has the largest area. • 2 — `Merge with neighboring polygon with identical attribute value, if any, or leave as is`: merges the small polygon geometry with a neighboring polygon that has the same attribute value, if such a polygon exists.

continues on next page

Table 24.138 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Field to consider when merging polygons with the identical attribute method Optional	`MERGE_ATTRIBUT`	[tablefield: any]	Field in the input layer to consider when merging polygons with the identical attribute method.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Small polygons merged layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from merging small polygons	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from merging small polygons	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the small polygon. • `gc_errory`: the y coordinate of the centroid of the small polygon. • `gc_error`: the area of the small polygon. • `UNIQUE_ID` field: the unique ID of the input feature that is a small polygon. • `report`: a text field describing the fix applied or justifying the failure. • `error_fixed`: a boolean field indicating whether the error was fixed.
Small polygons merged layer	`OUTPUT`	[vector: polygon]	Output layer of polygons with small polygons merged according to the chosen method. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryarea`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

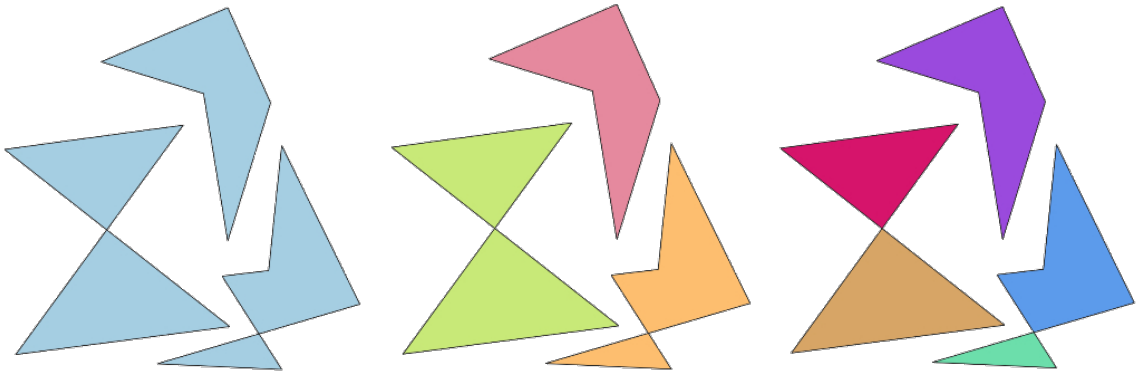
The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Split self-intersecting geometries

Splits self-intersecting geometries based on an error layer from the *Self-intersections* algorithm.

Pastaba

This algorithm detects invalid polygon geometries (self-intersections). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.



24.23 Fig. : Fixing self-intersection using various methods: as multi-part (middle), and multiple single-part geometries (right).

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm

continues on next page

Table 24.141 – tęsinys iš praeito puslapio

Label	Name	Type	Description
Method	METHOD	[enumeration]	Method to apply to the self-intersecting geometries: • 0 – Split feature into a multi-part feature: splits the self-intersecting geometry into multiple parts stored in a single multipart feature. • 1 – Split feature into multiple single-part features: splits the self-intersecting geometry into multiple parts, each part stored in a separate feature.
Field of original feature unique identifier	UNIQUE_ID	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm
Field of part index	PART_IDX	[tablefield: integer] Default: gc_partidx	Field storing the erroneous feature's geometry part number.
Field of ring index	RING_IDX	[tablefield: integer] Default: gc_ringidx	Field storing the erroneous feature's geometry ring number.
Field of vertex index	VERTEX_IDX	[tablefield: integer] Default: gc_vertidx	Field storing the erroneous feature's geometry vertex number.
Field of segment 1	SEGMENT_1	[tablefield: integer] Default: gc_segment_1	Field storing the number of the first segment involved in the erroneous intersection.
Field of segment 2	SEGMENT_2	[tablefield: integer] Default: gc_segment_2	Field storing the number of the second segment involved in the erroneous intersection.
Self-intersections fixed layer	OUTPUT	[vector: same as input] Default: [Create temporary layer]	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing self-intersections	REPORT	[vector: point] Default: [Create temporary layer]	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing self-intersections	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part where the self-intersection occurs. • `gc_ringidx`: the index of the feature's geometry ring where the self-intersection occurs. • `gc_vertidx` • `gc_errorx`: the x coordinate of the self-intersection. • `gc_errory`: the y coordinate of the self-intersection. • `gc_error` • `gc_segment_1`: the index of the first segment involved in the intersection. • `gc_segment_2`: the index of the second segment involved in the intersection. • `UNIQUE_ID` field: the unique ID of the self-intersecting input feature. • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed.
Self-intersections fixed layer	`OUTPUT`	[vector: same as input]	Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryselfintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

24.1.7 GPS

Keisti GPS duomenis

Naudoja GPSBabel įrankį įvairių GPS duomenų failų keitimui į GPX standarto formatą.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties failas	`INPUT`	[failas]	Failas su keičiamais duomenimis
Formatas	`FORMAT`	[sąrašas]	Keičiamo failo formatas, iš šio sąrašo.
Geoobjektų tipas	`FEATURE_TYPE`	[sąrašas] Numatytas: 0	Keičiamų duomenų tipas • 0 — Kelio taškai • 1 — Maršrutai • 2 — Pėdsakai
Išvestis	`OUTPUT`	[vector: any] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties GPX failio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	`OUTPUT_LAYER`	[vector: any]	Išvesties sluoksnis su GPX standarto formato duomenimis

Pythono kodas

Algoritmo ID: `native:convertgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Convert GPX feature type

Uses the GPSBabel tool to convert GPX features from one type to another (e.g. converting all waypoint features to a route).

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties failas	`INPUT`	[failas]	Failas su keičiamais duomenimis
Conversion	`CONVERSION`	[sąrašas] Numatytas: 0	The type of conversion to apply • 0 — Waypoints from a route • 1 — Waypoints from a track • 2 — Routes from waypoints • 3 — Tracks from waypoints
Išvestis	`OUTPUT`	[vector: point or line] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties failo specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvestis	`OUTPUT`	[vector: any]	Output layer with converted GPX features

Pythono kodas

Algorithm ID: `native:convertgpxfeaturetype`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Download GPS data from device

Uses the GPSBabel tool to download data from a GPS device into the GPX standard format.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Device	`DEVICE`	[sąrašas] Default:*Garmin serial*	The GPS device used to create the data. Must be declared in *GPS Settings* dialog.
Port	`PORT`	[sąrašas]	The port the device is connected to. Available ports depend on the OS.
Geoobjektų tipas	`FEATURE_TYPE`	[sąrašas] Numatytas: 0	Keičiamų duomenų tipas • 0 — Kelio taškai • 1 — Maršrutai • 2 — Pėdsakai
Išvestis	`OUTPUT`	[vector: any] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties failo specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	`OUTPUT_LAYER`	[vector: any]	Išvesties sluoksnis su GPX standarto formato duomenimis

Pythono kodas

Algorithm ID: `native:downloadgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Upload GPS data to device

Uses the GPSBabel tool to upload data to a GPS device from the GPX standard format.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties failas	`INPUT`	[failas]	`.GPX` file containing the data to upload
Device	`DEVICE`	[sąrašas] Default:*Garmin serial*	The GPS device you would like to upload the data to. Must be declared in *GPS Settings* dialog.
Port	`PORT`	[sąrašas]	The port the device is connected to. Available ports depend on the OS.
Geoobjektų tipas	`FEATURE_TYPE`	[sąrašas] Numatytas: 0	The type of data to upload • 0 — Kelio taškai • 1 — Maršrutai • 2 — Pėdsakai

Išvestys

No output is provided. If successful, data are loaded to the device.

Pythono kodas

Algorithm ID: `native:uploadgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.8 Interpoliacija

Intensyvumo žemėlapis (branduolio tankio įvertinimas)

Sukuria įvesties vektorinio taškų sluoksnio tankio (intensyvumo) rastrą naudojant branduolio tankio įvertinimą.

Tankis skaičiuojamas pagal taškų skaičių vietoje, kur didesni susispietusių taškų skaičiai reiškia didesnes reikšmes. Intensyvumo žemėlapiai leidžia lengvai identifikuoti taškų susispietimo „karštus taškus“.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Taškų sluoksnis	`INPUT`	[vektorius: taškai]	Vektorinis taškų sluoksnis, kurį reikia naudoti intensyvumo žemėlapiui
Spindulys	`RADIUS`	[skaičius: slankaus kablelio] Numatytas: 100.0	Intensyvumo žemėlapio paieškos spindulys (ar branduolio pralaidumas) žemėlapio vienetais. Spindulys nurodo atstumą aplinką tašką, kuriuo bus juntama taško įtaka. Didesnės reikšmės reikš didesnį glotninimą, o mažesnės reikšmės rodys mažesnes taškų tankumo detales.
Išvesties rastro dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 0.1	Išvesties rastro sluoksnio pikselio dydis sluoksnio vienetais. Naudotojo sąsajoje dydį galima nurodyti kaip skaičių eilučių (`Eilučių skaičius`) / stulpelių (`Stulpelių skaičius`) **ar** pikselio dydį (`Pikselio dydis X` / `Pikselio dydis Y`). Didinat eilučių ar stulpelių skaičių, mažės celės dydis ir didės išvesties rastro failo dydis. Atitinkamai bus keičiamos reikšmės `Eilutės`, `Stulpeliai`, `Pikselio dydis X` ir ``Pikselio dydis Y` - dvigubinant eilučių skaičių dvigubės ir stulpeliai, o celės dydis bus padalintas per pusę. Išvesties rastro apimtis liks tokia pati (daugmaž).
Spindulys iš lauko Pasirinktinis	`RADIUS_FIELD`	[lentelės laukas: skaičius]	Nustato kiekvieno geoobjekto paieškos spindulį pagal įvesties sluoksnio atributų lauką.
Svoris iš lauko Pasirinktinis	`WEIGHT_FIELD`	[lentelės laukas: skaičius]	Leidžia įvesties geoobjektams taikyti svorį iš atributų lauko. Tai gali būti naudojama didinat kai kurių geoobjektų įtaką gaunamame intensyvumo žemėlapyje.
Branduolio forma	`KERNEL`	[sąrašas] Numatyta: *0*	Valdo greitį, kuriuo taško įtaka mažėja didėjant atstumui nuo taško. Skirtingi branduoliai silpnėja skirtingais greičiais, taigi trigubo svorio branduolys reikš geoobjektus su didesniu svoriu atstumams arčiau taško nei Epanechnikovo branduolys. Tai reiškia, kad trigubo svorio branduolys sukurs „tikslesnius“ karštus taškus nei Epanechnikovo „glotnesni“ karštieji taškai. Yra daug skirtingų formų (daugiau informacijos rasite Wikipedia puslapyje): • 0 — Ketvirtinis • 1 — Trikampis • 2 — Vienodas • 3 — Trigubo svorio • 4 — Epanechnikovo

continues on next page

Table 24.151 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Mažėjimo greitis (tik trikampiams branduoliams) Pasirinktinis	`DECAY`	[skaičius: slankaus kablelio] Numatytas: *0.0*	Gali būti naudojamas su trikampiais branduoliais, norint dar labiau valdyti, kaip geoobjekto intensyvumas mažėja tolstant nuo geoobjekto. • Reikšmė 0 (=minimali) rodo, kad intensyvumas bus koncentruojamas nurodyto spindulio viduryje ir bus visiškai išnykęs kraštuose. • Reikšmė 0.5 nurodo, kad pikseliai spindulio krašte bus per pusę mažesnio intensyvumo nei nurodyto paieškos spindulio centre. • Reikšmė 1 reiškia, kad intensyvumas per visą paieškos spindulį paskirstytas tolygiai. (Tai analogiška branduoliui „Tolygus“.) • Už 1 didesnė reikšmė nurodo, kad intensyvumas didesnis ties paieškos spindulio kraštu, lyginant su centru.
Išvesties reikšmių mastelio keitimas	`OUTPUT_VALUE`	[sąrašas] Numatyta: *0*	Leidžia keisti išvesties intensyvumo rastro reikšmes. Viena iš: • 0 — Pradinis • 1 — Pakeistas
Intensyvumo žemėlapis	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite išvesties rastro sluoksnį su branduolio tankio reikšmėmis. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

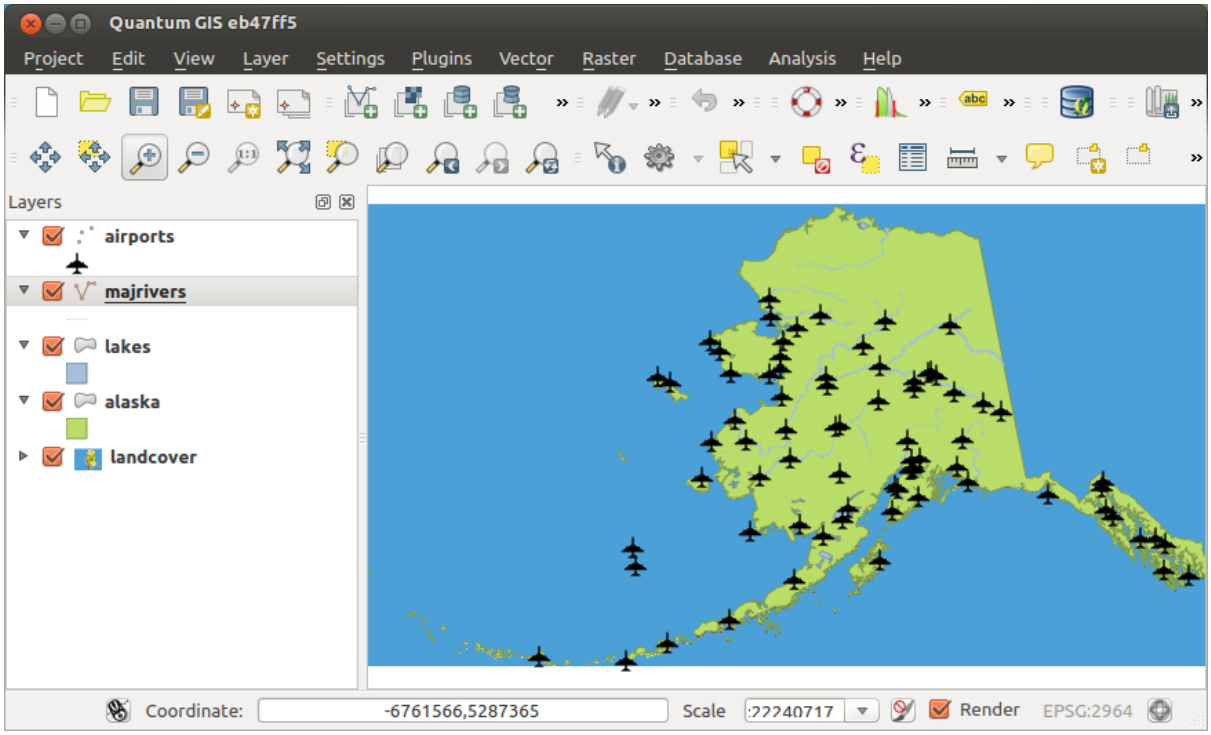
Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Intensyvumo žemėlapis	`OUTPUT`	[rastras]	Rastro sluoksnis su branduolio tankio reikšmėmis

Pavyzdys: intensyvumo žemėlapio kūrimas

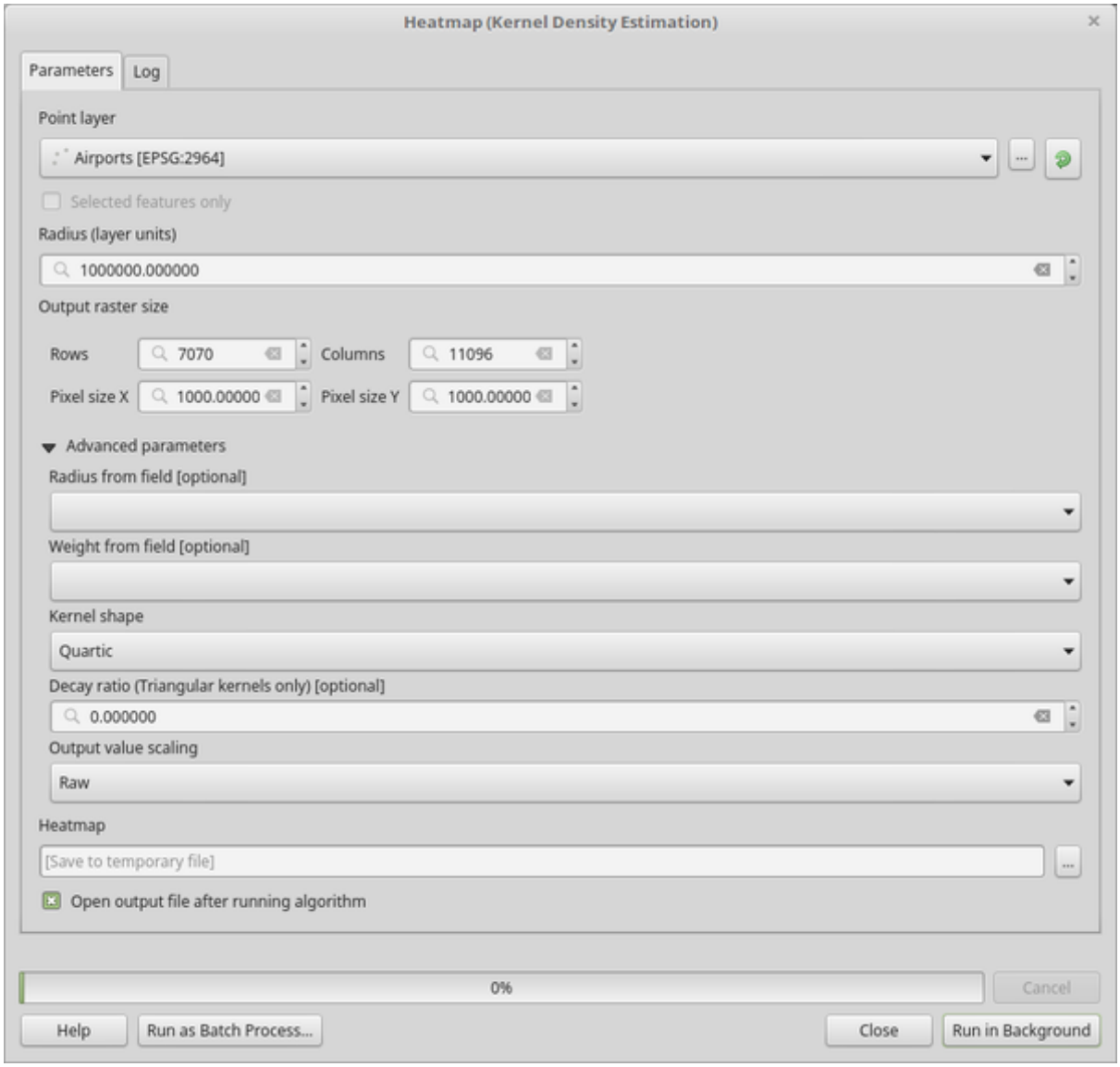
Šis pavyzdys naudos vektorinį taškų sluoksnį `airports` iš QGIS pavyzdinio duomenų rinkinio (žr. *Pavyzdinių duomenų atsisiuntimas*). Kitą puikią intensyvumo žemėlapių kūrimo QGIS pamoką galima rasti http://qgistutorials.com.

Fig. 24.24, rodomi Aliaskos oro uostai.



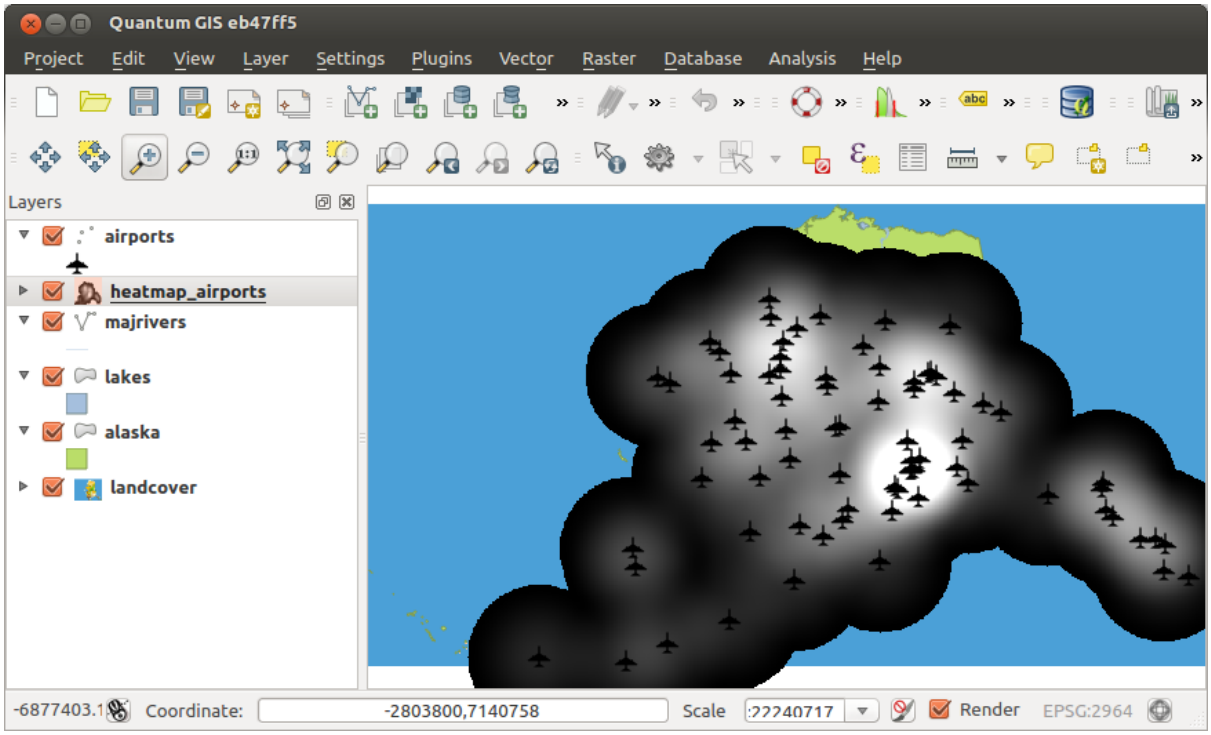
24.24 Fig. : Aliaskos oro uostai

1. Atverkite algoritmą *Intensyvumas (branduolio tankio įvertinimas)* iš QGIS grupės *Interpoliacija*
2. Lauke *Taškų sluoksnis* parinkite iš dabartiniame projekte įkeltų taškų sluoksnių sąrašo `airports`.
3. Pakeiskite *Spindulį* į `1000000` metrų.
4. Pakeiskite *Taško dydis X* į `1000`. *Taško dydis Y*, *Eilutės* ir *Stulpeliai* bus automatiškai pakeisti.
5. Spauskite *Vykdyti*, kad sukurtumėte ir įkeltumėte oro uostų intensyvumo žemėlapį (žr. Fig. 24.26).



24.25 Fig. : Intensyvumo žemėlapio dialogas

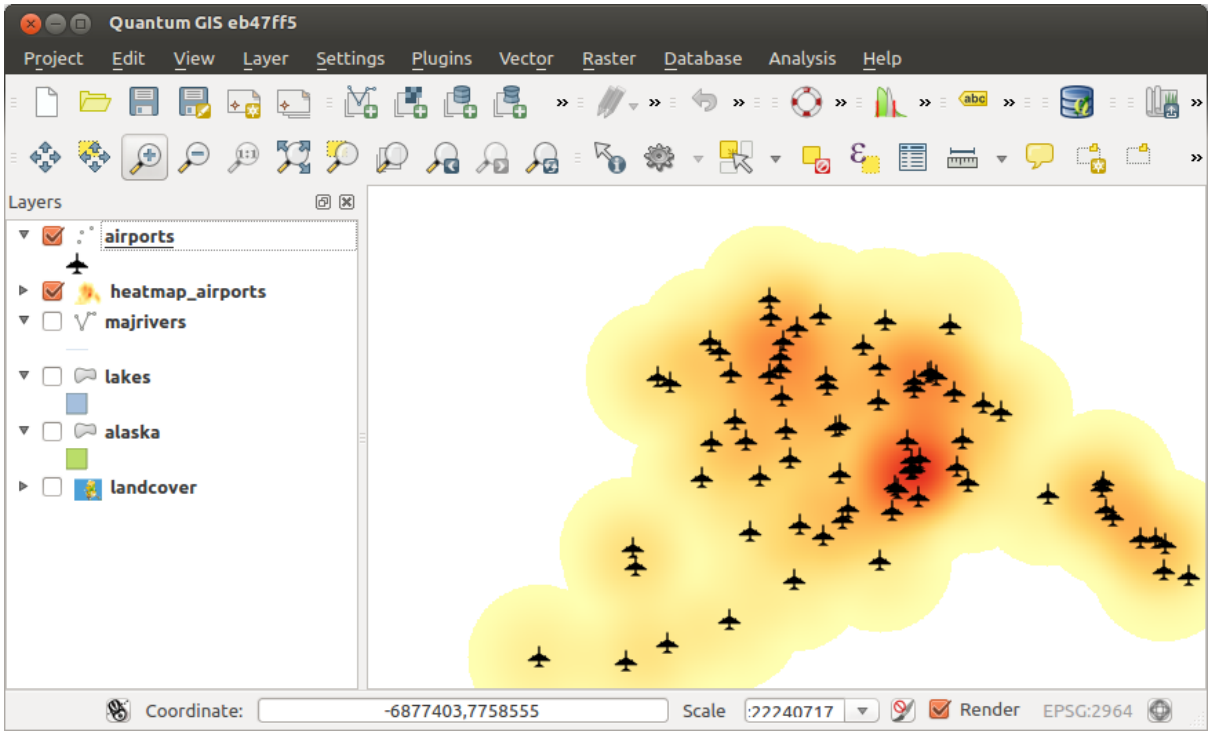
QGIS sukurs intensyvumo žemėlapį ir pridės jį į jūsų žemėlapio langą. Pagal nutylėjimą intensyvumo žemėlapis yra juodai baltas, kur šviesesnės zonos rodo didesnę oro uostų koncentraciją. Intensyvumo žemėlapio stilių dabar galima pakeisti, kad pagerintumėte jo išvaizdą.



24.26 Fig. : Po įkėlimo intensyvumo žemėlapis atrodo kaip pilkas paviršius

1. Atverkite sluoksnio `heatmap_airports` savybių dialogą (parinkite sluoksnį `heatmap_airports`, atidarykite kontekstinį meniu su dešiniu pelės mygtuku ir spauskite *Savybės*).
2. Pereikite į kortelę *Simbologija*.
3. Pakeiskite *Braižymo tipą* į ‚Vienos juostos pseudospalva‘.
4. Parinkite tinkamą *Spalvų rampą* , pavyzdžiui `YlOrRd`.
5. Spauskite mygtuką *Klasifikuoti*.
6. Spauskite *Gerai*, kad atnaujintumėte sluoksnį.

Galutinis rezultatas matomas Fig. 24.27.



24.27 Fig. : Stilizuotas Aliaskos oro uostų intensyvumo žemėlapis

Pythono kodas

Algoritmo ID: `qgis:heatmapkerneldensityestimation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Atvirkštinio atstumo svorio (IDW) interpoliacija

Kuria vektorinio taškų sluoksnio atvirkštinio atstumo svorio (IDW) interpoliaciją.

Pavyzdiniams taškams svoris priskiriamas taip interpoliuojant, kad vieno taško įtaką kitam mažėja su atstumu nuo nežinomo taško, kurį norite sukurti.

IDW interpoliacijos būdas taipogi turi kelis trūkumus: interpoliacijos rezultato kokybė gali kristi, jei turimų taškų duomenų pasiskirstymas yra netolygus.

Taipogi, maksimalios ir minimalios reikšmės interpoliuotame paviršiuje galis pasirodyti tik pradinių taškų vietose.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis(iai)	`INTERPOLATION_`	[tekstas]	Vektorinis sluoksnis(iai) ir laukas(ai), kuriuos reikia naudoti interpoliacijai, užkoduoti į tekstą (daugiau informacijos rasite klasės `ParameterInterpolationData` šaltinyje InterpolationWidgets). Šie GUI elementai teikiami interpoliacijos duomenų tekstui kurti: • **Vektorinis sluoksnis** [vektorius: bet koks] • **Interpoliacijos atributas** [lentelės laukas: skaitinis]: Interpoliacijai naudojamas atributas • **Interpoliacijai naudoti Z-koordinatę** [loginis]: Naudoja sluoksnyje įrašytas Z reikšmes (Numatyta: False) Kiekvienai pridėtai sluoksnio-lauko kombinacijai, galima pasirinkti tipą: • *Points*: Elevation sample locations containing the values to be interpolated (e.g., terrain height, Z). These points are the primary input for calculating the output surface. • *Structured lines*: Lines representing known continuous surface features that guide the terrain shape or structure (e.g., ridges, road centerlines, levees). They describe terrain direction while assuming continuity along the line. • *Break lines*: Lines representing known surface discontinuities or hard boundaries (e.g., cliffs, river banks, excavation edges, lake/pond outlines). They mark where the surface may change sharply. It is required if there are significant changes in elevation along the line. Sluoksnio-laiko elementai tekste atskiriami `'::|::'`. Sluoksnio-lauko sub-elementai atskiriami `'::~::'`.
Atstumo koeficientas P	`DISTANCE_COEFF`	[skaičius: slankaus kablelio] Numatyta: 2.0	Nustato interpoliacijos atstumo koeficientą. Minimalus: 0.0, maksimalus: 100.0.

continues on next page

Table 24.153 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Apimtis (xmin, xmax, ymin, ymax)	`EXTENT`	[apimtis]	Išvesties rastro sluoksnio apimtis. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Išvesties rastro dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 0.1	Išvesties rastro sluoksnio pikselio dydis sluoksnio vienetais. Naudotojo sąsajoje dydį galima nurodyti kaip skaičių eilučių (`Eilučių skaičius`) / stulpelių (`Stulpelių skaičius`) **ar** pikselio dydį (`Pikselio dydis X` / `Pikselio dydis Y`). Didinat eilučių ar stulpelių skaičių, mažės celės dydis ir didės išvesties rastro failo dydis. Atitinkamai bus keičiamos reikšmės `Eilutės`, `Stulpeliai`, `Pikselio dydis X` ir ``Pikselio dydis Y` - dvigubinant eilučių skaičių dvigubės ir stulpeliai, o celės dydis bus padalintas per pusę. Išvesties rastro apimtis liks tokia pati (daugmaž).
Interpoliuotas	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Rastro sluoksnis su interpoliuotomis reikšmėmis. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Interpoliuotas	`OUTPUT`	[rastras]	Interpoliuotų reikšmių rastro sluoksnis

Pythono kodas

Algoritmo ID: `qgis:idwinterpolation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

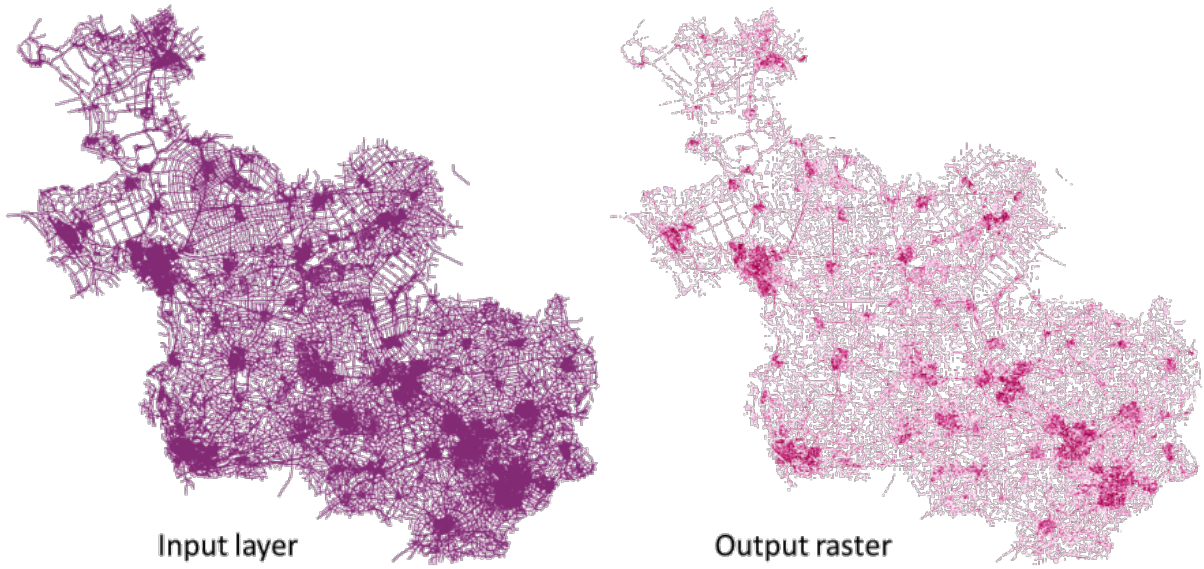
algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Linijų tankis

Kiekvienai rastro celei paskaičiuoja linijinių geoobjektų tankio matą apskritoje kaimynystėje. Šis matas gaunamas sumuojant visus linijų segmentus, kurie kertasi su apskritimine kaimynyste ir padalinant šią sumą iš tokios kaimynystės ploto. Linijų segmentams galima taikyti svorio faktorių.

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.



24.28 Fig. : Linijų tankio pavyzdys. Įvesties sluoksnio šaltinis: Roads Overijssel - Nyderlandai (OSM).

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties linijų sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis sluoksnis su linijų geoobjektais
Svorio laukas	`WEIGHT`	[lentelės laukas: skaičius]	Sluoksnio laukas, kuriame yra skaičiavime naudojamas svorio koeficientas

continues on next page

Table 24.155 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Paieškos spindulys	`RADIUS`	[skaičius: slankaus kablelio] Numatytas: 10.0	Apskritiminės kaimynystės spindulys. Čia galima nurodyti vienetus.
Taško dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 10.0	Išvesties rastro sluoksnio pikselio dydis sluoksnio vienetais. Rastras turi kvadratinius pikselius.
Linijų tankio rastras	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvestis kaip rastro sluoksnis. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Linijų tankio rastras	`OUTPUT`	[rastras]	Išvesties linijų tankio rastro sluoksnis.

Pythono kodas

Algoritmo ID: `native:linedensity`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

TIN interpoliacija

Kuria Trianguliuoto Netaisyklingo Tinklo (TIN) interpoliaciją iš vektorinio taškų sluoksnio.

Naudodami TIN metodą jūs galite sukurti paviršių, suformuotą iš artimiausių kaimyninių taškų trikampių. Kad tai būtų padaryta, aplink parinktus taškus sukuriami apskritimai, o jų susikirtimai sujungiami į tinklą nepersidengiančių ir kiek galima kompaktiškesnių trikampių. Gauti paviršiai nėra tolydūs.

Šis algoritmas sukuria tiek interpoliuotų reikšmių rastro sluoksnį, tiek ir trianguliuotų ribų vektorinį linijų sluoksnį.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis(iai)	`INTERPOLATION_`	[tekstas]	Vektorinis sluoksnis(iai) ir laukas(ai), kuriuos reikia naudoti interpoliacijai, užkoduoti į tekstą (daugiau informacijos rasite klasės `ParameterInterpolationData` šaltinyje InterpolationWidgets). Šie GUI elementai teikiami interpoliacijos duomenų tekstui kurti: • **Vektorinis sluoksnis** [vektorius: bet koks] • **Interpoliacijos atributas** [lentelės laukas: skaitinis]: Interpoliacijai naudojamas atributas • **Interpoliacijai naudoti Z-koordinatę** [loginis]: Naudoja sluoksnyje įrašytas Z reikšmes (Numatyta: False) Kiekvienai pridėtai sluoksnio-lauko kombinacijai, galima pasirinkti tipą: • *Points*: Elevation sample locations containing the values to be interpolated (e.g., terrain height, Z). These points are the primary input for calculating the output surface. • *Structured lines*: Lines representing known continuous surface features that guide the terrain shape or structure (e.g., ridges, road centerlines, levees). They describe terrain direction while assuming continuity along the line. • *Break lines*: Lines representing known surface discontinuities or hard boundaries (e.g., cliffs, river banks, excavation edges, lake/pond outlines). They mark where the surface may change sharply. It is required if there are significant changes in elevation along the line. Sluoksnio-laiko elementai tekste atskiriami `'::|::'`. Sluoksnio-lauko sub-elementai atskiriami `'::~::'`.
Interpoliacijos metodas	`METHOD`	[sąrašas] Numatytas: 0	Nustatykite naudojamą interpoliacijos metodą. Vienas iš: • *Tiesinis* • *Clough-Toucher (kubinis)*

continues on next page

Table 24.158 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Apimtis (xmin, xmax, ymin, ymax)	`EXTENT`	[apimtis]	Išvesties rastro sluoksnio apimtis. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Išvesties rastro dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 0.1	Išvesties rastro sluoksnio pikselio dydis sluoksnio vienetais. Naudotojo sąsajoje dydį galima nurodyti kaip skaičių eilučių (`Eilučių skaičius`) / stulpelių (`Stulpelių skaičius`) **ar** pikselio dydį (`Pikselio dydis X` / `Pikselio dydis Y`). Didinat eilučių ar stulpelių skaičių, mažės celės dydis ir didės išvesties rastro failo dydis. Atitinkamai bus keičiamos reikšmės `Eilutės`, `Stulpeliai`, `Pikselio dydis X` ir ``Pikselio dydis Y` - dvigubinant eilučių skaičių dvigubės ir stulpeliai, o celės dydis bus padalintas per pusę. Išvesties rastro apimtis liks tokia pati (daugmaž).
Interpoliuotas	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties TIN interpoliacija kaip rastro sluoksnis. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…
Trianguliacija	`TRIANGULATION`	[vektorius: linija] Numatytas: `[Praleisti išvestį]`	Išvesties TIN kaip vektorinis sluoksnis. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Interpoliuotas	`OUTPUT`	[rastras]	Išvesties TIN interpoliacija kaip rastro sluoksnis
Trianguliacija	`TRIANGULATION`	[vektorius: linija]	Išvesties TIN kaip vektorinis sluoksnis.

Pythono kodas

Algoritmo ID: `qgis:tininterpolation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.9 Sluoksnio įrankiai

Eksportuoti sluoksnio(ių) informaciją

Sukuria poligonų sluoksnį su geoobjektais, atitinkančiais parinkto sluoksnio(ių) apimtį.

Prie kiekvieno geoobjekto pridedama papildoma sluoksnio informacija (CRS, tiekėjo pavadinimas, failo kelias, sluoksnio pavadinimas, poaibio filtras, aprašymas ir autorių teisės).

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`LAYERS`	[sluoksnis] [sąrašas]	Įvesties sluoksniai, apie kuriuos reikia gauti informaciją.
Išvestis	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Išvesties sluoksnio su informacija specifikacija. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvestis	`OUTPUT`	[vektorius: poligonas]	Vektorinis poligonų sluoksnis, rodantis įvesties sluoksnių apimtį ir susijusią informaciją atributuose.

Pythono kodas

Algoritmo ID: `native:exportlayersinformation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Eksportuoti į skaičiuoklę

Eksportuoja vektorinių sluoksnių parinkimo atributus į skaičiuoklės dokumentą arba pasirinktinai prideda juos į esamą skaičiuoklę kaip papildomus lapus.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`LAYERS`	[vektorius: bet koks] [sąrašas]	Įvesties vektoriniai sluoksniai. Išvesties skaičiuoklėje kiekvienam sluoksniui bus sukurtas lapas su to sluoksnio atributais.
Naudoti laukų pseudonimus kaip antraštes	`USE_ALIAS`	[boolean] Numatytas: ne	Naudoti skaičiuoklei laukų pseudonimus iš atributų lentelės.
Eksportuoti formatuotas reikšmes vietoje pradinių	`FORMATTED_VALU`	[boolean] Numatytas: ne	Jei `True`, eksportuoja į skaičiuoklę formatuotas, žmogui skaitomas reikšmes (pavyzdžiui iš *reikšmių masyvo ar reikšmių ryšio*).
Perrašyti esamą skaičiuoklę	`OVERWRITE`	[boolean] Numatytas: Taip	Jei nurodyta skaičiuoklė jau yra, nustačius šioje parinktyje `True`, esama skaičiuoklė bus perrašyta. Jei ši parinktis yra `False`, o skaičiuoklė jau yra, sluoksniai bus pridėti kaip papildomi lapai.
Paskirties skaičiuoklė	`OUTPUT`	[file] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties skaičiuoklė su lapu kiekvienam sluoksniui. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Paskirties skaičiuoklė	`OUTPUT`	[file]	Skaičiuoklė su lapu kiekvienam sluoksniui.
Sluoksniai skaičiuoklėje	`OUTPUT_LAYERS`	[sąrašas]	Sąrašas sluoksnių, pridėtų į skaičiuoklę.

Pythono kodas

Algoritmo ID: `native:exporttospreadsheet`

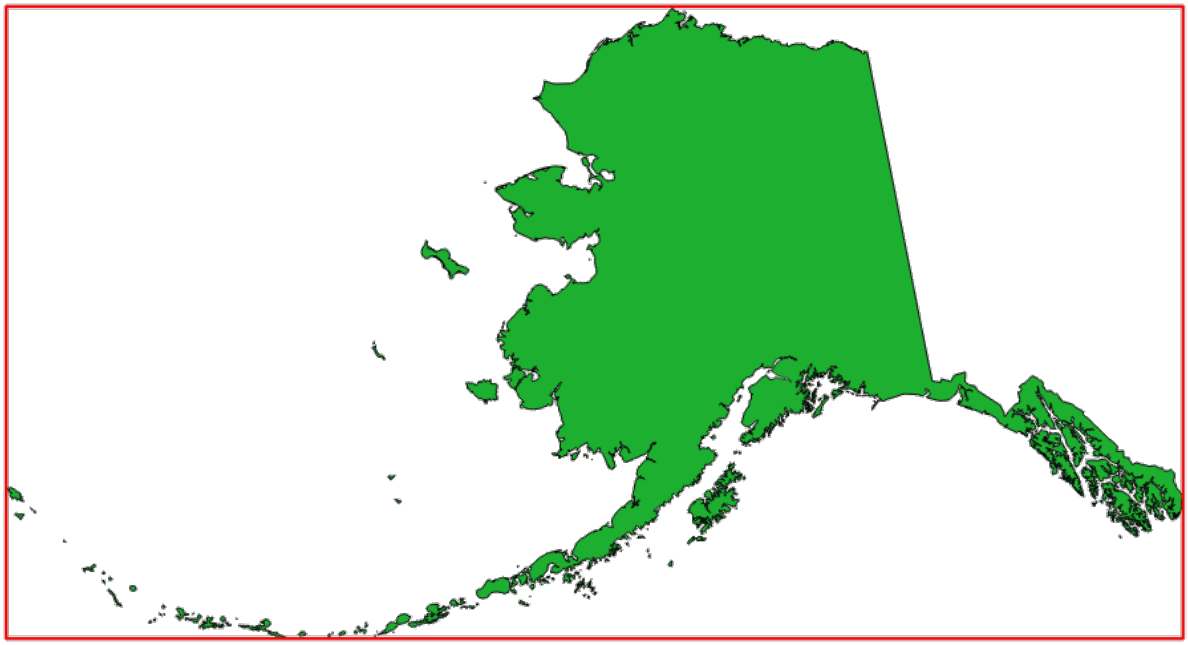
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Extract layer extent

Generates a vector layer with the minimum bounding box (rectangle with N-S orientation) that covers all the input features.

The output layer contains a single bounding box for the whole input layer.



24.29 Fig. : In red the bounding box of the source layer

Numatytas meniu: *Vektorius* ► *Tyrimo įrankiai*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Layer	`INPUT`	[layer]	Input layer
Apimtis	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the polygon vector layer for the output extent. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Apimtis	`OUTPUT`	[vektorius: poligonas]	Output (polygon) vector layer with the extent (minimum bounding box)

Pythono kodas

Algorithm ID: `native:polygonfromlayerextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.10 Tinklelis

Eksportuoti kontūrus

Sukuria kontūrus iš tinklelių skaliarinio duomenų rinkinio kaip vektorinį sluoksnį.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties tinklelio sluoksnis	`INPUT`	[tinklelis]	Tinklelio sluoksnis, iš kurio reikia eksportuoti duomenis
Duomenų aibių grupės	`DATASET_GROUPS`	[sluoksnis] [sąrašas]	Duomenų aibių grupės

continues on next page

Table 24.166 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Duomenų aibės laikas	DATASET_TIME	[datalaikas]	Laiko diapazonas, į kurį reikia atsižvelgti • 0 — Dabartinis drobės laikas • 1 — Nurodyta data/laikas • 2 — Duomenų aibės grupės laiko žingsnis
Žingsnis tarp kontūro lygių Pasirinktinis	INCREMENT	[skaičius: slankaus kablelio] Numatytas: jokio	Intervalas tarp sukurtų lygių.
Minimalus kontūro lygis Pasirinktinis	MINIMUM	[skaičius: slankaus kablelio] Numatytas: jokio	Pradinis kontūro lygis. Žemiau šios reikšmės nebus sukurtas joks lygis.
Maksimalus kontūro lygis Pasirinktinis	MAXIMUM	[skaičius: slankaus kablelio] Numatytas: jokio	Maksimalus kontūro lygis, t.y. aukščiau už šitą reikšmę nebus sukurtas jos kontūro lygis.
Kontūro lygių sąrašas Pasirinktinis	CONTOUR_LEVEL_	[tekstas] Numatytas: jokio	Norimų kontūro lygių sąrašas (atskirtas kableliais). Jei užpildytas, žingsnio, minimalaus ir maksimalaus lygio laukai bus ignoruojami.
Išvesties koordinačių sistema Pasirinktinis	CRS_OUTPUT	[crs]	Koordinačių atskaitos sistema, kurią priskirti išvesčiai
Eksportuotos kontūro linijos	OUTPUT_LINES	[vektorius: linija] Numatytas: [Kurti laikiną sluoksnį]	Nurodykite išvesties linijų sluoksnį, reprezentuojantį tinklelio sluoksnio kontūrus. *Vienas iš*: • Kurti laikiną sluoksnį (TEMPORARY_OUTPUT) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Eksportuoti kontūrų poligonai	OUTPUT_POLYGON	[vektorius: poligonas] Numatytas: [Kurti laikiną sluoksnį]	Nurodykite išvesties poligonų sluoksnį, reprezentuojantį tinklelio sluoksnio kontūrus. *Vienas iš*: • Kurti laikiną sluoksnį (TEMPORARY_OUTPUT) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Eksportuotos kontūro linijos	OUTPUT_LINES	[vektorius: linija]	Linijų sluoksnis su tinklelio sluoksnio kontūrais.
Eksportuoti kontūrų poligonai	OUTPUT_POLYGON	[vektorius: poligonas]	Poligonų sluoksnis su tinklelio sluoksnio kontūrais.

Pythono kodas

Algoritmo ID: `native:meshcontours`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Eksportuoti skerspjūvio duomenų aibės reikšmes tinklelio linijose

Ištraukia tinklelio duomenų aibės reikšmes iš linijų, esančių vektoriniame sluoksnyje.

Kiekviena linija diskretizuojama raiškos atstumo parametru ištraukiant reikšmes iš jos viršūnių.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties tinklelio sluoksnis	`INPUT`	[tinklelis]	Tinklelio sluoksnis, iš kurio reikia eksportuoti duomenis
Duomenų aibių grupės	`DATASET_GROUPS`	[sluoksnis] [sąrašas]	Duomenų aibių grupės
Duomenų aibės laikas	`DATASET_TIME`	[datalaikas]	Laiko diapazonas, į kurį reikia atsižvelgti • 0 — Dabartinis drobės laikas • 1 — Nurodyta data/laikas • 2 — Duomenų aibės grupės laiko žingsnis
Linijos duomenų eksportui	`INPUT_LINES`	[vektorius: linija]	Linijos, kur duomenys bus ištraukti iš duomenų aibės tinklelio
Linijų segmentavimo raiška	`RESOLUTION`	[skaičius: slankaus kablelio] Numatytas: 10.0	Atstumas tarp linijos taškų, kur duomenys bus ištraukti iš duomenų aibės tinklelio. Galima parinkti vienetus.
Skaitmenų skaičius koordinatėms	`COORDINATES_DI`	[skaičius: sveikas skaičius] Numatyta: 2	Skaičius skaitmenų koordinačių reikšmių apvalinimui
Duomenų aibės reikšmių skaitmenų skaičius	`DATASET_DIGITS`	[skaičius: sveikas skaičius] Numatyta: 2	Duomenų aibės reikšmių apvalinimo skaitmenų skaičius
CSV failas su eksportuotais duomenimis	`OUTPUT`	[file] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties failo specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
CSV failas su eksportuotais duomenimis	`OUTPUT`	[file]	

Pythono kodas

Algoritmo ID: `native:meshexportcrosssection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export mesh edges

Exports a mesh layer's edges to a line vector layer, with the dataset values on edges as attribute values.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties tinklelio sluoksnis	`INPUT`	[tinklelis]	Tinklelio sluoksnis, iš kurio reikia eksportuoti duomenis
Duomenų aibių grupės	`DATASET_GROUPS`	[sluoksnis] [sąrašas]	Duomenų aibių grupės
Duomenų aibės laikas	`DATASET_TIME`	[datalaikas]	Laiko diapazonas, į kurį reikia atsižvelgti • 0 — Dabartinis drobės laikas • 1 — Nurodyta data/laikas • 2 — Duomenų aibės grupės laiko žingsnis
Išvesties koordinačių sistema Pasirinktinis	`CRS_OUTPUT`	[crs]	Koordinačių atskaitos sistema, kurią priskirti išvesčiai
Export vector option	`VECTOR_OPTION`	[sąrašas]	Coordinate type of vector value exportation. • 0 — Cartesian (x,y) • 1 — Polar (magnitude, degree) • 2 — Cartesian and polar

continues on next page

Table 24.170 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Output vector layer	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Išvesties failo specifikacija. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Output vector layer	`OUTPUT`	[vektorius: linija]	Output vector line layer containing the edges of the input mesh layer with associated dataset values

Pythono kodas

Algorithm ID: `native:exportmeshedges`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export mesh faces

Exports a mesh layer's faces to a polygon vector layer, with the dataset values on faces as attribute values.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties tinklelio sluoksnis	`INPUT`	[tinklelis]	Tinklelio sluoksnis, iš kurio reikia eksportuoti duomenis
Duomenų aibių grupės	`DATASET_GROUPS`	[sluoksnis] [sąrašas]	Duomenų aibių grupės
Duomenų aibės laikas	`DATASET_TIME`	[datalaikas]	Laiko diapazonas, į kurį reikia atsižvelgti • 0 — Dabartinis drobės laikas • 1 — Nurodyta data/laikas • 2 — Duomenų aibės grupės laiko žingsnis
Išvesties koordinačių sistema Pasirinktinis	`CRS_OUTPUT`	[crs]	Koordinačių atskaitos sistema, kurią priskirti išvesčiai

continues on next page

Table 24.172 – tęsinys iš praeito puslapio

Užrašas		Pavadinimas	Tipas	Aprašymas
Export option	**vector**	`VECTOR_OPTION`	[sąrašas]	Coordinate type of vector value exportation. • 0 — Cartesian (x,y) • 1 — Polar (magnitude, degree) • 2 — Cartesian and polar
Output layer	**vector**	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Išvesties failo specifikacija. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas		Pavadinimas	Tipas	Aprašymas
Output layer	**vector**	`OUTPUT`	[vektorius: poligonas]	Output vector polygon layer containing the faces of the input mesh layer with associated dataset values

Pythono kodas

Algorithm ID: `native:exportmeshfaces`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export mesh on grid

Exports a mesh layer's dataset values to a gridded point vector layer, with the dataset values on this point as attribute values.

For data on volume (3D stacked dataset values), the exported dataset values are averaged on faces using the method defined in *the mesh layer properties* (default is Multi level averaging method). 1D meshes are not supported.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties tinklelio sluoksnis	`INPUT`	[tinklelis]	Tinklelio sluoksnis, iš kurio reikia eksportuoti duomenis
Duomenų aibių grupės	`DATASET_GROUPS`	[sluoksnis] [sąrašas]	Duomenų aibių grupės
Duomenų aibės laikas	`DATASET_TIME`	[datalaikas]	Laiko diapazonas, į kurį reikia atsižvelgti • 0 — Dabartinis drobės laikas • 1 — Nurodyta data/laikas • 2 — Duomenų aibės grupės laiko žingsnis
Apimtis Pasirinktinis	`EXTENT`	[apimtis]	Specify the spatial extent on which to process the data. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Grid spacing Pasirinktinis	`GRID_SPACING`	[skaičius: slankaus kablelio] Numatytas: 10.0	Spacing between the sample points to use
Išvesties koordinačių sistema Pasirinktinis	`CRS_OUTPUT`	[crs]	Koordinačių atskaitos sistema, kurią priskirti išvesčiai
Export vector option	`VECTOR_OPTION`	[sąrašas]	Coordinate type of vector value exportation. • 0 — Cartesian (x,y) • 1 — Polar (magnitude, degree) • 2 — Cartesian and polar
Output vector layer	`OUTPUT`	[vector: point] Numatytas: `[Kurti laikiną sluoksnį]`	Išvesties failo specifikacija. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Output vector layer	`OUTPUT`	[vector: point]	Output vector point layer with dataset values computed from the overlaid face.

Pythono kodas

Algorithm ID: `native:exportmeshongrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export mesh vertices

Exports a mesh layer's vertices to a point vector layer, with the dataset values on vertices as attribute values.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties tinklelio sluoksnis	`INPUT`	[tinklelis]	Tinklelio sluoksnis, iš kurio reikia eksportuoti duomenis
Duomenų aibių grupės	`DATASET_GROUPS`	[sluoksnis] [sąrašas]	Duomenų aibių grupės
Duomenų aibės laikas	`DATASET_TIME`	[datalaikas]	Laiko diapazonas, į kurį reikia atsižvelgti • 0 — Dabartinis drobės laikas • 1 — Nurodyta data/laikas • 2 — Duomenų aibės grupės laiko žingsnis
Išvesties koordinačių sistema Pasirinktinis	`CRS_OUTPUT`	[crs]	Koordinačių atskaitos sistema, kurią priskirti išvesčiai
Export vector option	`VECTOR_OPTION`	[sąrašas]	Coordinate type of vector value exportation. • 0 — Cartesian (x,y) • 1 — Polar (magnitude, degree) • 2 — Cartesian and polar
Output vector layer	`OUTPUT`	[vector: point] Numatytas: `[Kurti laikiną sluoksnį]`	Išvesties failo specifikacija. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Output vector layer	`OUTPUT`	[vector: point]	Output vector point layer containing the vertices of the input mesh layer with associated dataset values

Pythono kodas

Algorithm ID: `native:exportmeshvertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export time series values from points of a mesh dataset

Extracts a mesh dataset's time series values from points contained in a vector layer.

If the time step is kept to its default value (0 hours), the time step used is the one of the two first datasets of the first selected dataset group.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties tinklelio sluoksnis	`INPUT`	[tinklelis]	The mesh layer to extract data from
Duomenų aibių grupės	`DATASET_GROUPS`	[sluoksnis] [sąrašas]	Duomenų aibių grupės
Starting time	`STARTING_TIME`	[datalaikas]	The start of the time range to take into account • 0 — Dabartinis drobės laikas • 1 — Nurodyta data/laikas • 2 — Duomenų aibės grupės laiko žingsnis
Finishing time	`FINISHING_TIME`	[datalaikas]	The end of the time range to take into account • 0 — Dabartinis drobės laikas • 1 — Nurodyta data/laikas • 2 — Duomenų aibės grupės laiko žingsnis
Time step (hours) Pasirinktinis	`TIME_STEP`	[skaičius: slankaus kablelio] Numatytas: 0	Time between two consecutive steps to extract. Keep `0` to use time step of the first selected dataset group.
Points for data export	`INPUT_POINTS`	[vector: point]	Vector layer containing points where the data will be extracted from the dataset mesh

continues on next page

Table 24.178 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Skaitmenų skaičius koordinatėms	`COORDINATES_DI`	[skaičius: sveikas skaičius]	Number of digits for rounding coordinate values Numatyta: 2
Duomenų aibės reikšmių skaitmenų skaičius	`DATASET_DIGITS`	[skaičius: sveikas skaičius] Numatyta: 2	Number of digits for rounding dataset values
CSV failas su eksportuotais duomenimis	`OUTPUT`	[file] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties failo specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
CSV failas su eksportuotais duomenimis	`OUTPUT`	[file]	`.CSV` file containing the mesh dataset time series values at the overlaying point features

Pythono kodas

Algorithm ID: `native:meshexporttimeseries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Rasterize mesh dataset

Creates a raster layer from a mesh dataset.

For data on volume (3D stacked dataset values), the exported dataset values are averaged on faces using the method defined in *the mesh layer properties* (default is Multi level averaging method). 1D meshes are not supported.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties tinklelio sluoksnis	`INPUT`	[tinklelis]	Tinklelio sluoksnis, iš kurio reikia eksportuoti duomenis
Duomenų aibių grupės	`DATASET_GROUPS`	[sluoksnis] [sąrašas]	Duomenų aibių grupės

continues on next page

Table 24.180 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Duomenų aibės laikas	`DATASET_TIME`	[datalaikas]	Laiko diapazonas, į kurį reikia atsižvelgti • 0 — Dabartinis drobės laikas • 1 — Nurodyta data/laikas • 2 — Duomenų aibės grupės laiko žingsnis
Apimtis Pasirinktinis	`EXTENT`	[apimtis]	Specify the spatial extent on which to process the data. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Taško dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Pixel size of the output raster layer.
Išvesties koordinačių sistema Pasirinktinis	`CRS_OUTPUT`	[crs]	Koordinačių atskaitos sistema, kurią priskirti išvesčiai
Output raster layer	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties failo specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (`|`).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Output raster layer	`OUTPUT`	[rastras]	Output raster layer with dataset values computed from the mesh layer.

Pythono kodas

Algorithm ID: `native:meshrasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Surface to Polygon

Exports a polygon file containing the boundary of a mesh layer. The resulting polygon may contain holes and may be a multi-part polygon.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties tinklelio sluoksnis	`INPUT`	[tinklelis]	The input mesh layer whose boundary will be exported as a polygon
Išvesties koordinačių sistema Pasirinktinis	`OUTPUT_CRS`	[crs]	The coordinate reference system (CRS) for the output polygon layer. If not specified, the CRS of the input mesh layer will be used.
Output vector layer	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Išvesties failo specifikacija. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Output vector layer	`OUTPUT`	[vektorius: poligonas]	The resulting polygon layer containing the boundary of the mesh layer. The polygon may contain holes and may be multi-part.

Pythono kodas

Algorithm ID: `native:surfacetopolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

TIN mesh creation

Creates a TIN mesh layer from vector layers. The TIN mesh is created using a Delaunay triangulation.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`SOURCE_DATA`	[vektorius: bet koks] [sąrašas]	Vector layers to combine to generate the mesh layer
Vektorinis sluoksnis	GUI ONLY	[vektorius: bet koks] [sąrašas]	A selector for the vector layers to combine to generate the mesh layer
Value on vertex	GUI ONLY	[tablefield: any]	A selector of the field to use from the selected layer. Each vertex is assigned the corresponding value of its original feature.
Use Z-coordinate for value on vertex	GUI ONLY	[boolean] Numatytas: ne	If checked, the Z value of vector layer points or polygons/lines vertices will be used to assign the Z value of the vertex mesh layer. Only available if the input layers are 3D.
Output format	`MESH_FORMAT`	[sąrašas] Default: 2DM	Output format of the generated layer • 0 — 2DM • 1 — SELAFIN • 2 — PLY • 3 — Ugrid • 4 — Mike21
Išvesties koordinačių sistema Pasirinktinis	`CRS_OUTPUT`	[crs]	Koordinačių atskaitos sistema, kurią priskirti išvesčiai
Išvesties failas	`OUTPUT_MESH`	[tinklelis] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties failo specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties failas	`OUTPUT_MESH`	[tinklelis]	Output mesh layer with dataset values computed from the vector layers.

Pythono kodas

Algorithm ID: `native:tinmeshcreation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.11 Metadata tools

Add history metadata

Adds a new history entry to the layer's metadata.

Parameters

Label	Name	Type	Description
Layer	`INPUT`	[layer]	The input layer (vector, raster, etc.) to which the history entry will be added.
History entry	`HISTORY`	[string]	The text to be appended as a new entry in the layer's history metadata. This will be added to any existing history entries.

Outputs

Label	Name	Type	Description
Output	`OUTPUT`	[same as input]	The resulting layer with the updated history in its **Metadata properties**.

Python code

Algorithm ID: `native:addhistorymetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Copy Layer Metadata

Copies metadata from a source layer to a target layer. Any existing metadata in the target layer will be replaced with the metadata from the source layer. This includes all metadata fields, such as history, abstract, and other properties.

Parameters

Label	Name	Type	Description
Source layer	`SOURCE`	[layer]	The layer from which metadata will be copied.
Target layer	`TARGET`	[layer]	The layer to which metadata will be pasted. Any existing metadata in this layer will be replaced.
Save metadata as default	`DEFAULT`	[boolean] Default: False	If checked, the metadata information will be saved with the layer, hence available by default in subsequent projects.

Outputs

Label	Name	Type	Description
Output layer	`OUTPUT`	[layer]	The target layer with the metadata replaced by the metadata from the source layer. This includes all metadata fields, such as history, abstract, and other properties.

Python code

Algorithm ID: `native:copylayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Export Layer Metadata

Exports the metadata of a layer to a QMD file.

Parameters

Label	Name	Type	Description
Layer	`INPUT`	[layer]	The layer whose metadata will be exported.
Output	`OUTPUT`	[file] Default: `[Save to temporary file]`	Specification of the output file. *One of*: • Save to a Temporary File • Save to File…

Outputs

Label	Name	Type	Description
Output file	`OUTPUT`	[file]	The `.qmd` file containing the exported metadata. This file can be used to import metadata into another layer.

Python code

Algorithm ID: `native:exportlayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Set Layer Metadata

Applies metadata to a layer from a `.qmd` file.

Parameters

Label	Name	Type	Description
Layer	`INPUT`	[layer]	The layer to which the metadata will be applied.
Metadata file	`METADATA`	[file]	The `.qmd` file containing the metadata to be applied.
Save metadata as default	`DEFAULT`	[boolean] Default: False	If checked, the metadata information will be saved with the layer, hence available by default in subsequent projects.

Outputs

Label	Name	Type	Description
Output layer	`OUTPUT`	[same as input]	The input layer with the metadata replaced by the metadata from the `.qmd` file.

Python code

Algorithm ID: `native:setlayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Set Metadata Fields

Sets various metadata fields for a layer.

Parameters

Label	Name	Type	Description
Layer	`INPUT`	[layer]	The layer whose metadata fields will be updated
Identifier Optional	`IDENTIFIER`	[string]	A unique identifier for the layer
Parent identifier Optional	`PARENT_IDENTIF`	[string]	The identifier of the parent resource, if applicable
Title Optional	`TITLE`	[string]	The title of the layer
Type Optional	`TYPE`	[string]	The type of data stored in the layer
Language Optional	`LANGUAGE`	[string]	The language of the metadata
Encoding Optional	`ENCODING`	[string]	The character encoding used in the metadata
Abstract Optional	`ABSTRACT`	[string]	A brief description or abstract of the layer
Coordinate reference system Optional	`CRS`	[crs]	The coordinate reference system of the layer
Fees Optional	`FEES`	[string]	Information about any fees associated with accessing the layer
Ignore empty fields	`IGNORE_EMPTY`	[boolean] Default: False	If checked, no update will be done to metadata fields that are not filled

Outputs

Label	Name	Type	Description
Output layer	`OUTPUT`	[same as input]	The input layer with the specified metadata fields updated.

Python code

Algorithm ID: `native:setmetadatafields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

Update Layer Metadata

Copies all non-empty metadata fields from a source layer to a target layer. Leaves empty input fields unchanged in the target.

Parameters

Label	Name	Type	Description
Source layer	`SOURCE`	[layer]	The layer from which non-empty metadata fields will be copied
Target layer	`TARGET`	[layer]	The layer to which non-empty metadata fields will be pasted. Empty fields in the source layer will not overwrite existing metadata in the target layer.

Outputs

Label	Name	Type	Description
Target layer	`OUTPUT`	[layer]	The target layer with updated metadata.

Python code

Algorithm ID: `native:updatelayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Using processing algorithms from the console* for details on how to run processing algorithms from the Python console.

24.1.12 Modeliuotojo įrankiai

Įspėjimas

Šiuos įrankius galima naudoti tik modelių dizaineryje. Jų nėra apdorojimo įrankinėje.

Skaičiuoti išraišką

Skaičiuoja QGIS išraiškos rezultatą ir panaikina poreikį naudoti modelyje tą pačią išraišką kelis kartus, kai tas pats rezultatas reikalingas daugiau nei vieną kartą. Taipogi tai atveria galimybes, kurios kitu atveju būtų neįmanomos. Pavyzdžiui, jūs galite kurti laiko žymos reikšmę vieną kartą ir naudoti ją modelyje kelis kartus. Jei laiko žyma būtų skaičiuojama kiekvieną kartą, reikšmės būtų skirtingos.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvestis	`INPUT`	[išraiška]	Skaičiuojama išraiška

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Reikšmė	`OUTPUT`	[Gauta reikšmė]	Paskaičiuota rezultato reikšmė. Išvesties duomenų tipas kis priklausomai nuo konkrečios algoritme naudojamos išraiškos.

Pythono kodas

Algoritmo ID: `native:calculateexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Conditional branch

Adds a conditional branch into a model, allowing parts of the model to be executed based on the result of an expression evaluation. Mostly by using tool dependencies to control the flow of a model.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Field	`BRANCH`	[tekstas]	Name of the condition
Field	`CONDITION`	[išraiška]	Expression to evaluate

Išvestys

None.

Pythono kodas

Algorithm ID: `native:condition`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Create directory

Creates a new directory on a file system. Directories will be created recursively, creating all required parent directories in order to construct the full specified directory path. No errors will be raised if the directory already exists.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Directory path	`PATH`	[tekstas]	Folder path to create

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvestis	`OUTPUT`	[aplankas]	Created folder

Pythono kodas

Algorithm ID: `native:createdirectory`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Feature filter

Filters features from the input layer and redirects them to one or several outputs. If you do not know about any attribute names that are common to all possible input layers, filtering is only possible on the feature geometry and general record mechanisms, such as `$id` and `uuid`.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vector: any]	Įvesties sluoksnis.
Outputs and filters (one or more)	`OUTPUT_<name of the filter>`	[toks pat, kaip įvestis]	The output layers with filters (as many as there are filters).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvestis (one or more)	`native:filter_ of filter>`	[toks pat, kaip įvestis]	The output layers with filtered features (as many as there are filters).

Pythono kodas

Algorithm ID: `native:filter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Filter layers by type

Filters layers by their type. Incoming layers will be directed to different outputs based on whether they are a vector or raster layer.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[layer]	Generic Map Layer

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Vector features Pasirinktinis	`VECTOR`	[vector]	A Vector Layer of the input, if compatible
Rastro sluoksnis Pasirinktinis	`RASTER`	[rastras]	A Raster Layer of the input, if compatible

Pythono kodas

Algorithm ID: `native:filterlayersbytype`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Load layer into project

Loads a layer to the current project.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Layer	`INPUT`	[layer]	Layer to load in the legend
Loaded layer name	`NAME`	[tekstas]	Name of the loaded layer

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Layer	`OUTPUT`	[toks pat, kaip įvestis]	The (renamed) loaded layer

Pythono kodas

Algorithm ID: `native:loadlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raise exception

Raises an exception and cancels a model's execution. The exception message can be customized, and optionally an expression based condition can be specified. If an expression condition is used, then the exception will only be raised if the expression result is true. A false result indicates that no exception will be raised, and the model execution can continue uninterrupted.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Message	`MESSAGE`	[tekstas]	Message to display
Condition Pasirinktinis	`CONDITION`	[išraiška]	Expression to evaluate if true

Išvestys

A message in the log panel.

Pythono kodas

Algorithm ID: `native:raiseexception`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raise message

Raises an information message in the log. The message can be customized, and optionally an expression based condition can be specified. If an expression condition is used, then the message will only be logged if the expression result is true. A false result indicates that no message will be logged.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Information message	`MESSAGE`	[tekstas]	Message to display
Condition Pasirinktinis	`CONDITION`	[išraiška]	Expression to evaluate if true

Išvestys

A message in the log panel.

Pythono kodas

Algorithm ID: `native:raisemessage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raise warning

Raises a warning message in the log. The warning message can be customized, and optionally an expression based condition can be specified. If an expression condition is used, then the warning will only be logged if the expression result is true. A false result indicates that no warning will be logged.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Message	`MESSAGE`	[tekstas]	Message to display
Condition Pasirinktinis	`CONDITION`	[išraiška]	Expression to evaluate if true

Išvestys

A message in the log panel.

Pythono kodas

Algorithm ID: `native:raisewarning`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Rename layer

Renames a layer.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Layer	`INPUT`	[layer]	Layer to rename
New name	`NAME`	[tekstas]	The new name of the layer

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Layer	`OUTPUT`	[toks pat, kaip įvestis]	The (renamed) output layer

Pythono kodas

Algorithm ID: `native:renamelayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Save log to file

Saves the model's execution log to a file. Optionally, the log can be saved in a HTML formatted version.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Use HTML	`USE_HTML`	[Boolean] Numatytas: ne	Use HTML formatting

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
File	`OUTPUT`	[tekstas]	Destination of the log

Pythono kodas

Algorithm ID: `native:savelog`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Set project variable

Sets an expression variable for the current project.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Variable name	`NAME`	[tekstas]	Name of the variable
Variable value	`VALUE`	[tekstas]	Value to be stored

Išvestys

None.

Pythono kodas

Algorithm ID: `native:setprojectvariable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

String concatenation

Concatenates two strings into a single one in the Processing Modeler.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input 1	`INPUT_1`	[tekstas]	First string
Input 2	`INPUT_2`	[tekstas]	Second string

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Concatenation	`CONCATENATION`	[tekstas]	The concatenated string

Pythono kodas

Algorithm ID: `native:stringconcatenation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Variable distance buffer

Įspėjimas

This algorithm is deprecated and can be removed anytime. Prefer using *Buferis* algorithm instead.

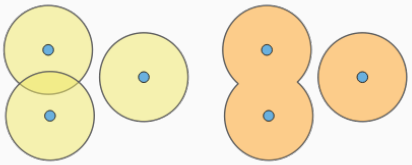
Computes a buffer area for all the features in an input layer.

The size of the buffer for a given feature is defined by an attribute, so it allows different features to have different buffer sizes.

Taip pat žiūrėkite

Buferis

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vector: any]	Įvesties vektorinis sluoksnis
Distance field	`DISTANCE`	[lentelės laukas: skaičius]	Attribute for the distance radius of the buffer
Segmentai	`SEGMENTS`	[skaičius: sveikas skaičius] Default: *5*	Valdo skaičių linijų segmentų, kurie naudojami aproksimuojant apskritimo ketvirčius kuriant užapvalintus poslinkius.
Sulieti rezultatą	`DISSOLVE`	[boolean] Default: *False*	Choose to dissolve the final buffer, resulting in a single feature covering all input features.  24.30 Fig. : Normal and dissolved buffer
Pabaigos stilius	`END_CAP_STYLE`	[sąrašas] Default: Round	Controls how line endings are handled in the buffer. 24.31 Fig. : Apvali, plokščia arba kvadratinė pabaiga
Jungimo stilius	`JOIN_STYLE`	[sąrašas] Default: Round	Specifies whether round, miter or beveled joins should be used when offsetting corners in a line.

continues on next page

Table 24.220 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Mitros riba	`MITER_LIMIT`	[skaičius: slankaus kablelio] Numatyta: 2.0	Only applicable for mitered join styles, and controls the maximum distance from the offset curve to use when creating a mitered join.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Buferis	`OUTPUT`	[vektorius: poligonas]	Buffer polygon vector layer.

Pythono kodas

Algorithm ID: `qgis:variabledistancebuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.13 Tinklo analizė

Aptarnavimo plotas (pagal sluoksnį)

Grąžina visas kraštines ar jų dalis tinklo, kurį galima pasiekti nurodytu atstumu ar per nurodytą laiką iš taškinio sluoksnio. Tai leidžia įvertinti pasiekiamumą tinkle, pavyzdžiui, į kokias vietas galima nuvykti kelių tinklu neišleidžiant daugiau nei nurodytą reikšmę (kaina gali būti atstumas arba laikas).

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Vektorinis sluoksnis reprezentuojantis tinklą	`INPUT`	[vektorius: linija]	Linijų vektorinis sluoksnis, reprezentuojantis padengiamą tinklą

continues on next page

Table 24.222 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Vektorinis sluoksnis su pradžios taškais	START_POINTS	[vektorius: taškas]	Taškų vektorinis sluoksnis, kurio geoobjektai turi būti naudojami kaip pradžios taškai kuriant aptarnavimo zonas
Skaičiuojamo kelio tipas	STRATEGY	[sąrašas] Numatytas: 0	Skaičiuojamo kelio tipas. Vienas iš: • 0 — Trumpiausias • 1 — Greičiausias
Kelionės kaina (atstumas „Trumpiausiam“, laikas „Greičiausiam“)	TRAVEL_COST	[skaičius: slankaus kablelio] Numatytas: 0.0	Reikšmė vertinama kaip atstumas (tinklo sluoksnio vienetais) kai ieškoma „Trumpiausio“ kelio ir kaip laikas (valandomis), kai ieškoma „Greičiausio“ kelio.
Aptarnavimo plotas (linijos)	OUTPUT_LINES	[vektorius: linija] Numatytas: [Kurti laikiną sluoksnį]	Nurodykite išvesties linijų sluoksnį aptarnavimo plotui. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (TEMPORARY_OUTPUT) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Aptarnavimo plotas (ribų taškai)	OUTPUT	[vektorius: taškas] Numatytas: [Praleisti išvestį]	Nurodykite išvesties taškų sluoksnį aptarnavimo ploto ribiniams taškams. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (TEMPORARY_OUTPUT) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Nemaršrutizuojami geoobjektai	OUTPUT_NON_ROU	[vektorius: taškas] Numatytas: [Praleisti išvestį]	Nurodykite išvestį, kurioje bus saugomi bet kurie įvesties geoobjektai, iki kurių nepavyko numaršrutizuoti (pvz. kurie yra per toli nuo tinklo sluoksnio). *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (TEMPORARY_OUTPUT) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Krypties laukas Pasirinktinis	DIRECTION_FIEL	[lentelės laukas: bet koks]	Laukas, kuriame nurodomos tinklo kraštinių kryptys. Šio lauko reikšmės nurodomos su trimis parametrais Reikšmė krypčiai pirmyn, Reikšmė krypčiai atgal ir Reikšmė abiems kryptims. Kryptys pirmyn ir atgal nurodomos vienos krypties kraštinėms, „abiejų krypčių“ nurodomos abipusio eismo kraštinėms. Jei geoobjektas neturi reikšmės šiame lauke arba jei laukas nenurodytas, naudojama numatytoji krypties reikšmė (pateikiama su parametru Numatytoji kryptis parameter).
Reikšmė krypčiai pirmyn Pasirinktinis	VALUE_FORWARD	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti kraštines su eismu pirmyn.
Reikšmė krypčiai atgal Pasirinktinis	VALUE_BACKWARD	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti kraštines su eismu atgal.
Reikšmė abiems kryptims Pasirinktinis	VALUE_BOTH	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti abiejų krypčių eismą kraštinėje
Numatytoji kryptis	DEFAULT_DIRECT	[sąrašas] Numatyta: 2	Jei geoobjekto krypties lauke reikšmė nenurodyta arba laukas nenurodytas, tai naudojama ši reikšmė. Viena iš: • 0 — Kryptis pirmyn • 1 — Kryptis atgal • 2 — Abi kryptys
Greičio laukas Pasirinktinis	SPEED_FIELD	[lentelės laukas: skaičius]	Laukas, teikiantis greičio reikšmę (nurodoma km/h) tinklo kraštinėms, kai ieškoma greičiausio kelio. Jei šiame geoobjekto lauke reikšmė nenurodyta arba laukas nenurodytas, tai naudojama numatytoji greičio reikšmė (pateikta su parametru Numatytas greitis).
Numatytas greitis (km/h)	DEFAULT_SPEED	[skaičius: slankaus kablelio] Numatytas: 50.0	Kelionės laikui skaičiuoti naudojama reikšmė, jei nurodytame lauke nenurodyta greičio reikšmė
Topologijos tolerancija	TOLERANCE	[skaičius: slankaus kablelio] Numatytas: 0.0	Dvi linijos, kurių viršūnės yra arčiau viena kitos nei nurodytas tolerancijos atstumas, laikomos sujungtomis

Įtraukti viršutinės/apatinės ribos taškus	INCLUDE_BOUNDS	[loginis] Numatytas: ne	Sukuria taškų sluoksnio išvestį su dviem taškais kiekvienai kraštinei ties aptarnavimo ploto riba. Vienas taškas yra kraštinės pradžia, o kitas - pabaiga.
Maksimalus taško atstumas nuo tinklo Pasirinktinis	POINT_TOLERANC	[skaičius: slankaus kablelio] Numatytas: jokio	Nurodo neprivalomą taško atstumo nuo tinklo sluoksnio apribojimą. Jei taškas yra toliau nuo tinklo nei šis atstumas, jis bus laikomas nemaršrutizuojamu. Nenurodžius, pabaigos taškai bus pritraukti prie artimiausio taško tinklo sluoksnyje, nepriklausomai nuo to, kaip toli jie realiai yra nuo tinklo.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Aptarnavimo plotas (ribų taškai)	OUTPUT	[vektorius: taškas]	Išvesties taškų sluoksnis su aptarnavimo ploto ribų taškais.
Aptarnavimo plotas (linijos)	OUTPUT_LINES	[vektorius: linija]	Linijos sluoksnis, reprezentuojantis tinklo dalis, kurias gali aptarnauti pradžios taškai už nurodytą kainą.
Nemaršrutizuojami geoobjektai	OUTPUT_NON_ROU	[vektorius: taškas]	Neprivaloma išvestis, kurioje gali būti saugomi bet kokie įvesties geoobjektai, iki kurių nepavyko numaršrutizuoti (pvz. kurie yra per toli nuo tinklo sluoksnio).

Pythono kodas

Algorithm ID: `native:serviceareafromlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Aptarnavimo plotas (nuo taško)

Grąžina visas kraštines ar jų dalis tinklo, kurį galima pasiekti nurodytu atstumu ar per nurodytą laiką iš taškinio geoobjekto. Tai leidžia įvertinti pasiekiamumą tinkle, pavyzdžiui, į kokias vietas galima nuvykti kelių tinklu neišleidžiant daugiau nei nurodytą reikšmę (kaina gali būti atstumas arba laikas).

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Vektorinis sluoksnis reprezentuojantis tinklą	`INPUT`	[vektorius: linija]	Linijų vektorinis sluoksnis, reprezentuojantis padengiamą tinklą
Skaičiuojamo kelio tipas	`STRATEGY`	[sąrašas] Numatytas: 0	Skaičiuojamo kelio tipas. Vienas iš: • 0 — Trumpiausias • 1 — Greičiausias
Pradžios taškas	`START_POINT`	[koordinatės]	Koordinatės taško, aplink kurį reikia skaičiuoti aptarnavimo plotą. Spauskite greta parinkties esantį mygtuką … ir tada spauskite drobėje, kad užpildytumėte parametrą parinkto taško koordinatėmis.
Kelionės kaina (atstumas „Trumpiausiam“, laikas „Greičiausiam“)	`TRAVEL_COST`	[skaičius: slankaus kablelio] Numatytas: 0.0	Reikšmė vertinama kaip atstumas (tinklo sluoksnio vienetais) kai ieškoma „Trumpiausio“ kelio ir kaip laikas (valandomis), kai ieškoma „Greičiausio“ kelio.
Aptarnavimo plotas (linijos) Pasirinktinis	`OUTPUT_LINES`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties linijų sluoksnį aptarnavimo plotui. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Aptarnavimo plotas (ribų taškai) Pasirinktinis	`OUTPUT`	[vektorius: taškas] Numatytas: `[Praleisti išvestį]`	Nurodykite išvesties taškų sluoksnį aptarnavimo ploto ribiniams taškams. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Krypties laukas Pasirinktinis	DIRECTION_FIEL	[lentelės laukas: bet koks]	Laukas, kuriame nurodomos tinklo kraštinių kryptys. Šio lauko reikšmės nurodomos su trimis parametrais Reikšmė krypčiai pirmyn, Reikšmė krypčiai atgal ir Reikšmė abiems kryptims. Kryptys pirmyn ir atgal nurodomos vienos krypties kraštinėms, „abiejų krypčių“ nurodomos abipusio eismo kraštinėms. Jei geoobjektas neturi reikšmės šiame lauke arba jei laukas nenurodytas, naudojama numatytoji krypties reikšmė (pateikiama su parametru Numatytoji kryptis parameter).
Reikšmė krypčiai pirmyn Pasirinktinis	VALUE_FORWARD	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti kraštines su eismu pirmyn.
Reikšmė krypčiai atgal Pasirinktinis	VALUE_BACKWARD	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti kraštines su eismu atgal.
Reikšmė abiems kryptims Pasirinktinis	VALUE_BOTH	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti abiejų krypčių eismą kraštinėje
Numatytoji kryptis	DEFAULT_DIRECT	[sąrašas] Numatyta: 2	Jei geoobjekto krypties lauke reikšmė nenurodyta arba laukas nenurodytas, tai naudojama ši reikšmė. Viena iš: • 0 — Kryptis pirmyn • 1 — Kryptis atgal • 2 — Abi kryptys
Greičio laukas Pasirinktinis	SPEED_FIELD	[lentelės laukas: skaičius]	Laukas, teikiantis greičio reikšmę (nurodoma km/h) tinklo kraštinėms, kai ieškoma greičiausio kelio. Jei šiame geoobjekto lauke reikšmė nenurodyta arba laukas nenurodytas, tai naudojama numatytoji greičio reikšmė (pateikta su parametru Numatytas greitis).
Numatytas greitis (km/h)	DEFAULT_SPEED	[skaičius: slankaus kablelio] Numatytas: 50.0	Kelionės laikui skaičiuoti naudojama reikšmė, jei nurodytame lauke nenurodyta greičio reikšmė
Topologijos tolerancija	TOLERANCE	[skaičius: slankaus kablelio] Numatytas: 0.0	Dvi linijos, kurių viršūnės yra arčiau viena kitos nei nurodytas tolerancijos atstumas, laikomos sujungtomis

Maksimalus taško atstumas nuo tinklo Pasirinktinis	POINT_TOLERANC	[skaičius: slankaus kablelio] Numatytas: 0.0	Nurodo neprivalomą atstumą nuo pradžios taško iki tinklo sluoksnio. Jei taškas yra toliau nuo tinklo nei šis atstumas, bus registruojama klaida. Nenurodžius, taškas bus pritrauktas prie artimiausio taško tinklo sluoksnyje, nepriklausomai nuo to, kaip toli jis realiai yra nuo tinklo.
Įtraukti viršutinės/apatinės ribos taškus	INCLUDE_BOUNDS	[loginis] Numatytas: ne	Sukuria taškų sluoksnio išvestį su dviem taškais kiekvienai kraštinei ties aptarnavimo ploto riba. Vienas taškas yra kraštinės pradžia, o kitas - pabaiga.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Aptarnavimo plotas (ribų taškai)	OUTPUT	[vektorius: taškas]	Išvesties taškų sluoksnis su aptarnavimo ploto ribų taškais.
Aptarnavimo plotas (linijos)	OUTPUT_LINES	[vektorius: linija]	Linijos sluoksnis, reprezentuojantis tinklo dalis, kurias gali aptarnauti pradžios taškas už nurodytą kainą.

Pythono kodas

Algoritmo ID: `native:serviceareafrompoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Trumpiausias kelias (sluoksnis iki taško)

Skaičiuoja optimalius (trumpiausius ar greičiausius) kelius iš kelių pradinių taškų, apibrėžtų vektoriniame sluoksnyje, iki nurodyto pabaigos taško.

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Vektorinis sluoksnis reprezentuojantis tinklą	`INPUT`	[vektorius: linija]	Linijų vektorinis sluoksnis, reprezentuojantis padengiamą tinklą
Skaičiuojamo kelio tipas	`STRATEGY`	[sąrašas] Numatytas: 0	Skaičiuojamo kelio tipas. Vienas iš: • 0 — Trumpiausias • 1 — Greičiausias
Vektorinis sluoksnis su pradžios taškais	`START_POINTS`	[vektorius: taškas]	Taškų vektorinis sluoksnis, kurio geoobjektai naudojami kaip maršrutų pradžios taškai
Pabaigos taškas (x, y)	`END_POINT`	[koordinatės]	Taškinis objektas, reprezentuojantis maršrutų pabaigos tašką Spauskite greta parinkties esantį mygtuką … ir tada spauskite drobėje, kad užpildytumėte parametrą parinkto taško koordinatėmis.
Trumpiausias kelias	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties sluoksnį trumpiausiems keliams. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Nemaršrutizuojami geoobjektai Pasirinktinis	`OUTPUT_NON_ROU`	[vektorius: taškas] Numatytas: `[Praleisti išvestį]`	Nurodykite išvestį, kurioje bus saugomi bet kurie įvesties geoobjektai, iki kurių nepavyko numaršrutizuoti (pvz. kurie yra per toli nuo tinklo sluoksnio). *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Krypties laukas Pasirinktinis	DIRECTION_FIEL	[lentelės laukas: bet koks]	Laukas, kuriame nurodomos tinklo kraštinių kryptys. Šio lauko reikšmės nurodomos su trimis parametrais Reikšmė krypčiai pirmyn, Reikšmė krypčiai atgal ir Reikšmė abiems kryptims. Kryptys pirmyn ir atgal nurodomos vienos krypties kraštinėms, „abiejų krypčių“ nurodomos abipusio eismo kraštinėms. Jei geoobjektas neturi reikšmės šiame lauke arba jei laukas nenurodytas, naudojama numatytoji krypties reikšmė (pateikiama su parametru Numatytoji kryptis parameter).
Reikšmė krypčiai pirmyn Pasirinktinis	VALUE_FORWARD	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti kraštines su eismu pirmyn.
Reikšmė krypčiai atgal Pasirinktinis	VALUE_BACKWARD	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti kraštines su eismu atgal.
Reikšmė abiems kryptims Pasirinktinis	VALUE_BOTH	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti abiejų krypčių eismą kraštinėje
Numatytoji kryptis	DEFAULT_DIRECT	[sąrašas] Numatyta: 2	Jei geoobjekto krypties lauke reikšmė nenurodyta arba laukas nenurodytas, tai naudojama ši reikšmė. Viena iš: • 0 — Kryptis pirmyn • 1 — Kryptis atgal • 2 — Abi kryptys
Greičio laukas Pasirinktinis	SPEED_FIELD	[lentelės laukas: skaičius]	Laukas, teikiantis greičio reikšmę (nurodoma km/h) tinklo kraštinėms, kai ieškoma greičiausio kelio. Jei šiame geoobjekto lauke reikšmė nenurodyta arba laukas nenurodytas, tai naudojama numatytoji greičio reikšmė (pateikta su parametru Numatytas greitis).
Numatytas greitis (km/h)	DEFAULT_SPEED	[skaičius: slankaus kablelio] Numatytas: 50.0	Kelionės laikui skaičiuoti naudojama reikšmė, jei nurodytame lauke nenurodyta greičio reikšmė
Topologijos tolerancija	TOLERANCE	[skaičius: slankaus kablelio] Numatytas: 0.0	Dvi linijos, kurių viršūnės yra arčiau viena kitos nei nurodytas tolerancijos atstumas, laikomos sujungtomis

Maksimalus taško atstumas nuo tinklo Pasirinktinis	POINT_TOLERANC	[skaičius: slankaus kablelio] Numatytas: jokio	Nurodo neprivalomą atstumo tarp pradžios ir pabaigos taškų apribojimą tinklo sluoksniui. Jei pradžios geoobjektas yra toliau nuo tinklo nei šis atstumas, jis bus laikomas nemaršrutizuojamu. Jei pabaigos taškas yra toliau nuo tinklo nei šis atstumas, bus rodoma klaida. Nenurodžius taškai bus pritraukiami prie artimiausio taško tinklo sluoksnyje, nepriklausomai nuo to, kaip toli tinklo jie yra.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Trumpiausias kelias	`OUTPUT`	[vektorius: linija]	Trumpiausio ar greičiausio kelio nuo kiekvieno pradžios taško iki pabaigos taško linijų sluoksnis
Nemaršrutizuojami geoobjektai	`OUTPUT_NON_ROU`	[vektorius: taškas]	Neprivalomas išvesties sluoksnis, kuriame gali būti saugomi bet kokie įvesties geoobjektai, iki kurių nepavyko numaršrutizuoti (pvz. kurie yra per toli nuo tinklo sluoksnio).

Pythono kodas

Algoritmo ID: `native:shortestpathlayertopoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Trumpiausias kelias (taškas iki sluoksnio)

Skaičiuoja optimalius (trumpiausius arba greičiausius) kelius tarp nurodyto pradžios taško ir kelių pabaigos taškų, apibrėžtų taškų vektoriniame sluoksnyje.

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Vektorinis sluoksnis reprezentuojantis tinklą	`INPUT`	[vektorius: linija]	Linijų vektorinis sluoksnis, reprezentuojantis padengiamą tinklą

continues on next page

Table 24.231 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Skaičiuojamo kelio tipas	`STRATEGY`	[sąrašas] Numatytas: 0	Skaičiuojamo kelio tipas. Vienas iš: • 0 — Trumpiausias • 1 — Greičiausias
Pradžios taškas	`START_POINT`	[koordinatės]	Taškų geoobjektas, reprezentuojantis maršrutų pradžios tašką Spauskite greta parinkties esantį mygtuką … ir tada spauskite drobėje, kad užpildytumėte parametrą parinkto taško koordinatėmis.
Vektorinis sluoksnis su pabaigos taškais	`END_POINTS`	[vektorius: taškas]	Taškų vektorinis sluoksnis, kurio geoobjektai naudojami kaip maršrutų pabaigos taškai
Trumpiausias kelias	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties sluoksnį trumpiausiems keliams. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Nemaršrutizuojami geoobjektai	`OUTPUT_NON_ROU` Pasirinktinis	[vektorius: taškas] Numatytas: `[Praleisti išvestį]`	Nurodykite išvestį, kurioje bus saugomi bet kurie įvesties geoobjektai, iki kurių nepavyko numaršrutizuoti (pvz. kurie yra per toli nuo tinklo sluoksnio). *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Krypties laukas Pasirinktinis	`DIRECTION_FIEL`	[lentelės laukas: bet koks]	Laukas, kuriame nurodomos tinklo kraštinių kryptys. Šio lauko reikšmės nurodomos su trimis parametrais `Reikšmė krypčiai pirmyn`, `Reikšmė krypčiai atgal` ir `Reikšmė abiems kryptims`. Kryptys pirmyn ir atgal nurodomos vienos krypties kraštinėms, „abiejų krypčių“ nurodomos abipusio eismo kraštinėms. Jei geoobjektas neturi reikšmės šiame lauke arba jei laukas nenurodytas, naudojama numatytoji krypties reikšmė (pateikiama su parametru `Numatytoji kryptis` parameter).

continues on next page

Table 24.232 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Reikšmė krypčiai pirmyn Pasirinktinis	VALUE_FORWARD	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti kraštines su eismu pirmyn.
Reikšmė krypčiai atgal Pasirinktinis	VALUE_BACKWARD	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti kraštines su eismu atgal.
Reikšmė abiems kryptims Pasirinktinis	VALUE_BOTH	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti abiejų krypčių eismą kraštinėje
Numatytoji kryptis	DEFAULT_DIRECT	[sąrašas] Numatyta: 2	Jei geoobjekto krypties lauke reikšmė nenurodyta arba laukas nenurodytas, tai naudojama ši reikšmė. Viena iš: • 0 — Kryptis pirmyn • 1 — Kryptis atgal • 2 — Abi kryptys
Greičio laukas Pasirinktinis	SPEED_FIELD	[lentelės laukas: skaičius]	Laukas, teikiantis greičio reikšmę (nurodoma km/h) tinklo kraštinėms, kai ieškoma greičiausio kelio. Jei šiame geoobjekto lauke reikšmė nenurodyta arba laukas nenurodytas, tai naudojama numatytoji greičio reikšmė (pateikta su parametru Numatytas greitis).
Numatytas greitis (km/h)	DEFAULT_SPEED	[skaičius: slankaus kablelio] Numatytas: 50.0	Kelionės laikui skaičiuoti naudojama reikšmė, jei nurodytame lauke nenurodyta greičio reikšmė
Topologijos tolerancija	TOLERANCE	[skaičius: slankaus kablelio] Numatytas: 0.0	Dvi linijos, kurių viršūnės yra arčiau viena kitos nei nurodytas tolerancijos atstumas, laikomos sujungtomis
Maksimalus taško atstumas nuo tinklo Pasirinktinis	POINT_TOLERANC	[skaičius: slankaus kablelio] Numatytas: jokio	Nurodo neprivalomą atstumo tarp pradžios ir pabaigos taškų apribojimą tinklo sluoksniui. Jei pradžios taškas yra toliau nuo tinklo nei šis atstumas, bus rodoma klaida. Jei pabaigos geoobjektas yra toliau nuo tinklo nei šis atstumas, jis bus laikomas nemaršrutizuojamu. Nenurodžius taškai bus pritraukiami prie artimiausio taško tinklo sluoksnyje, nepriklausomai nuo to, kaip toli tinklo jie yra.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Trumpiausias kelias	OUTPUT	[vektorius: linija]	Trumpiausio ar greičiausio kelio nuo kiekvieno pradžios taško iki pabaigos taško linijų sluoksnis

continues on next page

Table 24.233 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Nemaršrutizuojami geoobjektai	`OUTPUT_NON_ROU`	[vektorius: taškas]	Neprivalomas išvesties sluoksnis, kuriame gali būti saugomi bet kokie įvesties geoobjektai, iki kurių nepavyko numaršrutizuoti (pvz. kurie yra per toli nuo tinklo sluoksnio).

Pythono kodas

Algoritmo ID: `native:shortestpathpointtolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Shortest path (point to point)

Computes the optimal (shortest or fastest) route between a given start point and a given end point.

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Vektorinis sluoksnis reprezentuojantis tinklą	`INPUT`	[vektorius: linija]	Linijų vektorinis sluoksnis, reprezentuojantis padengiamą tinklą
Skaičiuojamo kelio tipas	`STRATEGY`	[sąrašas] Numatytas: 0	Skaičiuojamo kelio tipas. Vienas iš: • 0 — Trumpiausias • 1 — Greičiausias
Start point (x, y)	`START_POINT`	[koordinatės]	Point feature representing the start point of the routes. Spauskite greta parinkties esantį mygtuką … ir tada spauskite drobėje, kad užpildytumėte parametrą parinkto taško koordinatėmis.

continues on next page

Table 24.234 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Pabaigos taškas (x, y)	`END_POINT`	[koordinatės]	Point feature representing the end point of the routes. Spauskite greta parinkties esantį mygtuką … ir tada spauskite drobėje, kad užpildytumėte parametrą parinkto taško koordinatėmis.
Trumpiausias kelias	`OUTPUT`	[vektorius: linija]	Nurodykite išvesties sluoksnį trumpiausiems keliams. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Krypties laukas Pasirinktinis	`DIRECTION_FIEL`	[lentelės laukas: bet koks]	Laukas, kuriame nurodomos tinklo kraštinių kryptys. Šio lauko reikšmės nurodomos su trimis parametrais `Reikšmė krypčiai pirmyn`, `Reikšmė krypčiai atgal` ir `Reikšmė abiems kryptims`. Kryptys pirmyn ir atgal nurodomos vienos krypties kraštinėms, „abiejų krypčių“ nurodomos abipusio eismo kraštinėms. Jei geoobjektas neturi reikšmės šiame lauke arba jei laukas nenurodytas, naudojama numatytoji krypties reikšmė (pateikiama su parametru `Numatytoji kryptis` parameter).
Reikšmė krypčiai pirmyn Pasirinktinis	`VALUE_FORWARD`	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti kraštines su eismu pirmyn.
Reikšmė krypčiai atgal Pasirinktinis	`VALUE_BACKWARD`	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti kraštines su eismu atgal.
Reikšmė abiems kryptims Pasirinktinis	`VALUE_BOTH`	[tekstas] Numatyta: ‚‘ (tuščias tekstas)	Reikšmė krypties lauke, identifikuojanti abiejų krypčių eismą kraštinėje
Numatytoji kryptis	`DEFAULT_DIRECT`	[sąrašas] Numatyta: 2	Jei geoobjekto krypties lauke reikšmė nenurodyta arba laukas nenurodytas, tai naudojama ši reikšmė. Viena iš: • 0 — Kryptis pirmyn • 1 — Kryptis atgal • 2 — Abi kryptys

continues on next page

Table 24.235 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Greičio laukas Pasirinktinis	`SPEED_FIELD`	[lentelės laukas: skaičius]	Laukas, teikiantis greičio reikšmę (nurodoma `km/h`) tinklo kraštinėms, kai ieškoma greičiausio kelio. Jei šiame geoobjekto lauke reikšmė nenurodyta arba laukas nenurodytas, tai naudojama numatytoji greičio reikšmė (pateikta su parametru `Numatytas greitis`).
Numatytas greitis (km/h)	`DEFAULT_SPEED`	[skaičius: slankaus kablelio] Numatytas: 50.0	Kelionės laikui skaičiuoti naudojama reikšmė, jei nurodytame lauke nenurodyta greičio reikšmė
Topologijos tolerancija	`TOLERANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Dvi linijos, kurių viršūnės yra arčiau viena kitos nei nurodytas tolerancijos atstumas, laikomos sujungtomis
Maksimalus taško atstumas nuo tinklo Pasirinktinis	`POINT_TOLERANC`	[skaičius: slankaus kablelio] Numatytas: jokio	Specifies an optional limit on the distance from the start and end points to the network layer. If either point is further from the network than this distance an error will be raised. If not set, points will be snapped to the nearest point on the network layer, regardless of how far away from the network they actually are.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Trumpiausias kelias	`OUTPUT`	[vektorius: linija]	Line layer of the shortest or fastest path from each of the start point to the end point

Pythono kodas

Algorithm ID: `native:shortestpathpointtopoint`

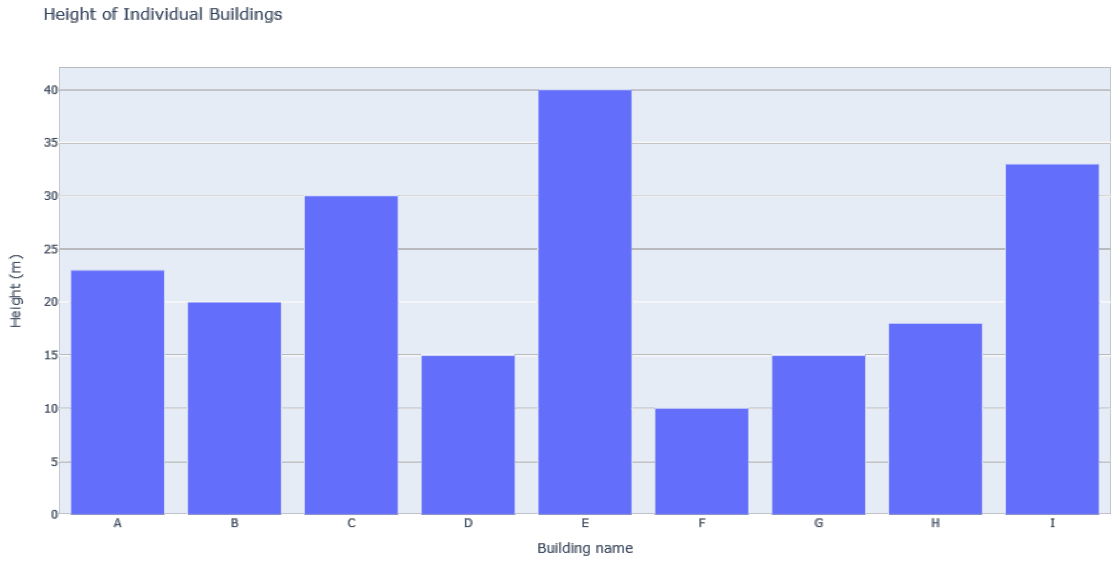
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.14 Diagramos

Stulpelinė diagrama

Sukuria stulpelinę diagramą iš kategorijos ir sluoksnio lauko.



24.32 Fig. : Bar plot example

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Kategorijos lauko pavadinimas	`NAME_FIELD`	[lentelės laukas: bet koks]	Kategorijos laukas, kuris naudojamas stulpelių grupavimui (X ašis)
Reikšmės laukas	`VALUE_FIELD`	[lentelės laukas: skaičius]	Reikšmė, kuri naudojama braižymui (Y ašis)
Title Optional	`TITLE`	[string] Default: „“	Title of the plot
X-axis Title Optional	`XAXIS_TITLE`	[string] Default: „“	If empty, the name of the category field is used. With a single space, the axis title is hidden.
Y-axis Title Optional	`YAXIS_TITLE`	[string] Default: „“	If empty, the name of the value field is used. With a single space, the axis title is hidden.
Stulpelinė diagrama	`OUTPUT`	[html] Numatytas: `[Įrašyti į laikiną failą]`	Įrašykite brėžinio HTML failą. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Stulpelinė diagrama	`OUTPUT`	[html]	HTML failas su brėžiniu. Galima rasti *Apdorojimas ► Rezultatų žiūryklė*.

Pythono kodas

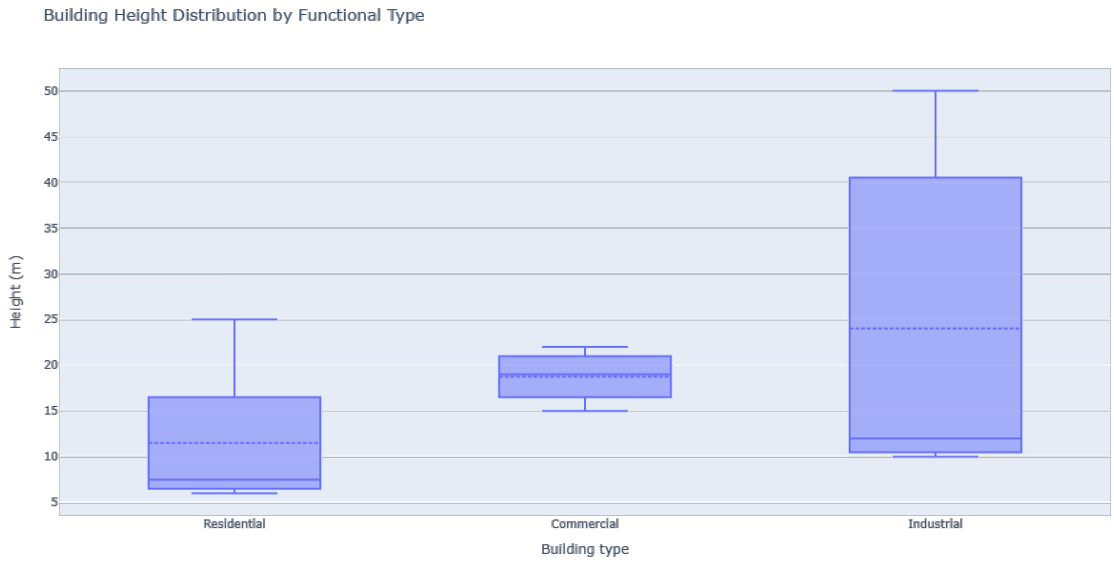
Algoritmo ID: `qgis:barplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Dėžučių diagrama

Sukuria dėžučių diagramą iš kategorijos ir skaičiaus sluoksnio lauko.



24.33 Fig. : Example of box plot

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Kategorijos pavadinimo laukas	`NAME_FIELD`	[lentelės laukas: bet koks]	Kategorijos laukas, kuris naudojamas dėžučių (X ašies) grupavimui
Reikšmės laukas	`VALUE_FIELD`	[lentelės laukas: skaičius]	Reikšmė, kuri naudojama braižymui (Y ašis)
Papildomos statistinės linijos	`MSD`	[sąrašas] Numatytas: 0	Į brėžinį pridedama papildoma statistinė informacija. Vienas iš: • 0 — Rodyti vidurkį • 1 — Rodyti standartinį nuokrypį • 2 — Nerodyti vidurkio ir standartinio nuokrypio
Title Optional	`TITLE`	[string] Default: „“	Title of the plot
X-axis Title Optional	`XAXIS_TITLE`	[string] Default: „“	If empty, the name of the category field is used. With a single space, the axis title is not shown.
Y-axis Title Optional	`YAXIS_TITLE`	[string] Default: „“	If empty, the name of the value field is used. With a single space, the axis title is not shown.
Dėžučių diagrama	`OUTPUT`	[html] Numatytas: `[Įrašyti į laikiną failą]`	Įrašykite brėžinio HTML failą. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Dėžučių diagrama	`OUTPUT`	[html]	HTML failas su brėžiniu. Galima rasti *Apdorojimas ► Rezultatų žiūryklė*.

Pythono kodas

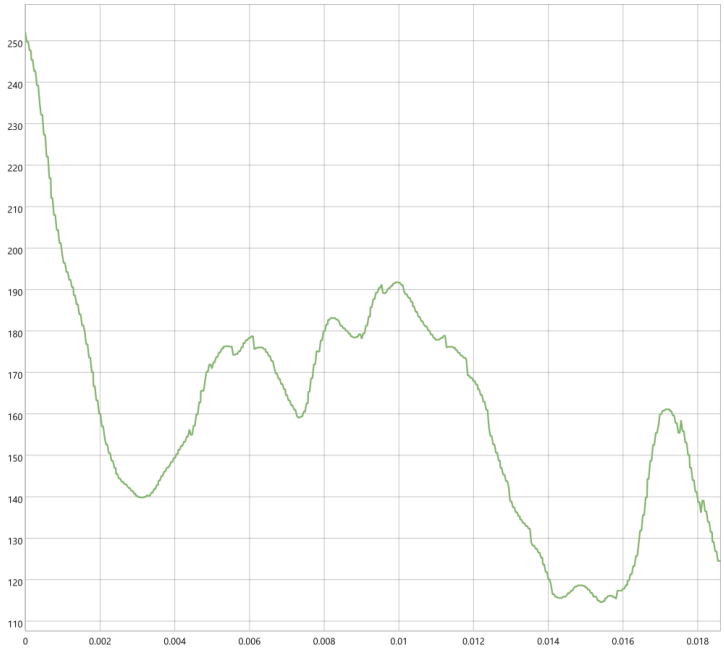
Algoritmo ID: `qgis:boxplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Generate elevation profile image

Creates an elevation profile image from a list of map layers and an optional terrain layer.



24.34 Fig. : Example of elevation profile

Parametrai

Basic parameters

Užrašas	Pavadinimas	Tipas	Aprašymas
Profile curve	`CURVE`	[geometry: line]	The LineString geometry (WKT or GeoJSON) representing the profile curve along which the elevation profile will be generated
Map layers	`MAP_LAYERS`	[layer] [list]	The list of map layers to include in the elevation profile
Chart width (in pixels)	`WIDTH`	[skaičius: sveikas skaičius] Default: 400 Minimum value: 0	The width of the output chart in pixels.
Chart height (in pixels)	`HEIGHT`	[skaičius: sveikas skaičius] Default: 300 Minimum value: 0	The height of the output chart in pixels.
Terrain layer Optional	`TERRAIN_LAYER`	[rastras]	A terrain layer (e.g., DEM) to use for elevation data. If not provided, elevation data will be derived from the map layers.

Advanced parameters

Užrašas	Pavadinimas	Tipas	Aprašymas
Chart minimum distance (X axis) Optional	MINIMUM_DISTAN	[numeric: double]	The minimum distance (X axis) to display on the chart. If not specified, the chart will auto-scale.
Chart maximum distance (X axis) Optional	MAXIMUM_DISTAN	[numeric: double]	The maximum distance (X axis) to display on the chart. If not specified, the chart will auto-scale.
Chart minimum elevation (Y axis) Optional	MINIMUM_ELEVAT	[numeric: double]	The minimum elevation (Y axis) to display on the chart. If not specified, the chart will auto-scale.
Chart maximum elevation (Y axis) Optional	MAXIMUM_ELEVAT	[numeric: double]	The maximum elevation (Y axis) to display on the chart. If not specified, the chart will auto-scale.
Chart text color Optional	TEXT_COLOR	[color]	The color of the text in the chart (e.g., axis labels, titles).
Chart background color Optional	BACKGROUND_COL	[color]	The background color of the chart.
Chart border color Optional	BORDER_COLOR	[color]	The color of the chart border.
Profile tolerance	TOLERANCE	[numeric: double] Default: 5.0 Minimum value: 0	Defines how far a feature (vector point, line, polygon, or point cloud) can be from the profile line to be included in the results. It uses map units and does not affect other layer types.
Chart DPI	DPI	[skaičius: sveikas skaičius] Default: 96 Minimum value: 0	The resolution of the output image in dots per inch (DPI).
Output image	OUTPUT	[file] Numatytas: [Įrašyti į laikiną failą]	Specify the image file for the plot. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Output image	OUTPUT	[file]	The generated elevation profile image in the specified format.

Pythono kodas

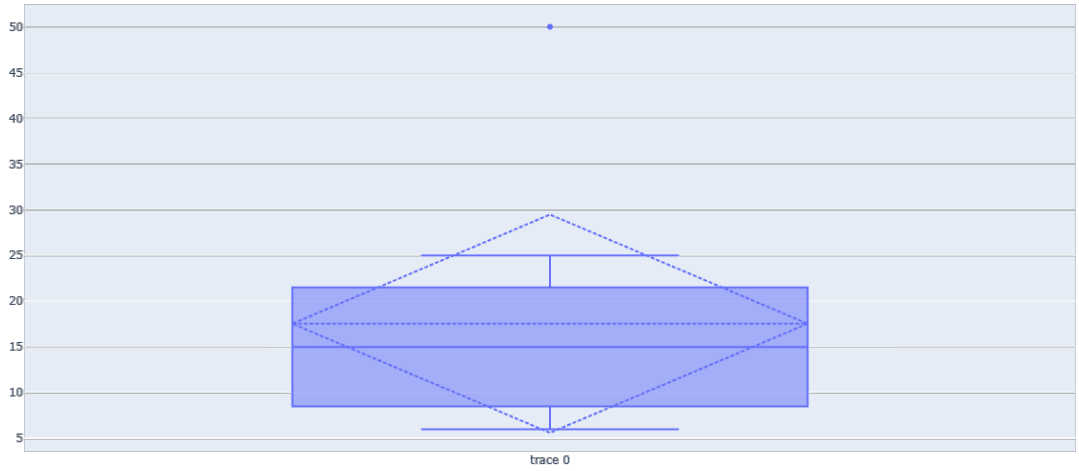
Algorithm ID: `native:generateelevationprofileimage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Vidurkio ir standartinio nuokrypio brėžinys

Sukuria brėžinį su vidurkio ir standartinio nuokrypio reikšmėmis.



24.35 Fig. : Example of mean and standard deviation plot

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties lentelė	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Kategorijos pavadinimo laukas	`NAME_FIELD`	[lentelės laukas: bet koks]	Kategorijos laukas, kuris naudojamas dėžučių (X ašies) grupavimui
Reikšmės laukas	`VALUE_FIELD`	[lentelės laukas: skaičius]	Reikšmė, kuri naudojama braižymui (Y ašis)
Brėžinys	`OUTPUT`	[html] Numatytas: `[Įrašyti į laikiną failą]`	Įrašykite brėžinio HTML failą. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Brėžinys	`OUTPUT`	[html]	HTML failas su brėžiniu. Galima rasti *Apdorojimas* ► *Rezultatų žiūryklė*.

Pythono kodas

Algoritmo ID: `qgis:meanandstandarddeviationplot`

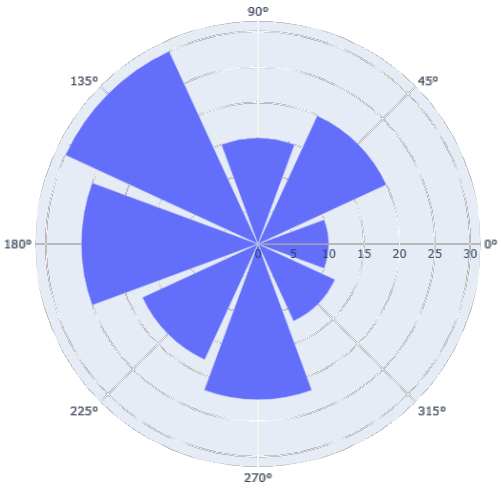
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Polar plot

Generates a polar plot based on the value of an input vector layer.

Two fields must be entered as parameters: one that defines the category each feature (to group features) and another one with the variable to plot (this has to be a numeric one).



24.36 Fig. : Example of polar plot

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Kategorijos pavadinimo laukas	`NAME_FIELD`	[lentelės laukas: bet koks]	Categorical field to use for grouping the features (X axis)
Reikšmės laukas	`VALUE_FIELD`	[lentelės laukas: skaičius]	Reikšmė, kuri naudojama braižymui (Y ašis)
Polar plot	`OUTPUT`	[html] Numatytas: `[Įrašyti į laikiną failą]`	Įrašykite brėžinio HTML failą. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Polar plot	`OUTPUT`	[html]	HTML failas su brėžiniu. Galima rasti *Apdorojimas* ► *Rezultatų žiūryklė*.

Pythono kodas

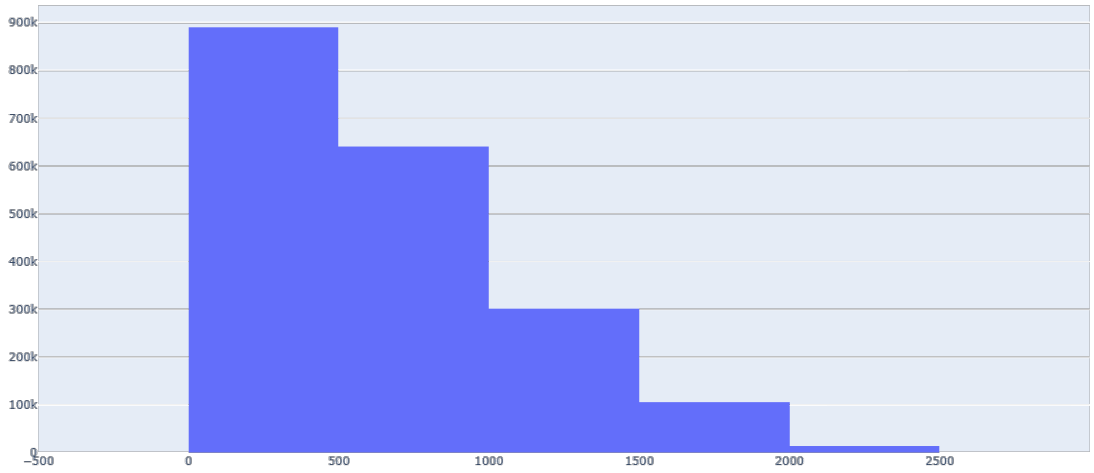
Algorithm ID: `qgis:polarplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster layer histogram

Generates a histogram with the values of a raster layer.



24.37 Fig. : Raster layer histogram example

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta]	Raster band to use for the histogram
number of bins	`BINS`	[skaičius: sveikas skaičius] Numatytas: 10	The number of bins to use in the histogram (X axis). Minimum 2.
Histogram	`OUTPUT`	[html] Numatytas: `[Įrašyti į laikiną failą]`	Įrašykite brėžinio HTML failą. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Histogram	`OUTPUT`	[html]	HTML failas su brėžiniu. Galima rasti *Apdorojimas ► Rezultatų žiūryklė*.

Pythono kodas

Algorithm ID: `qgis:rasterlayerhistogram`

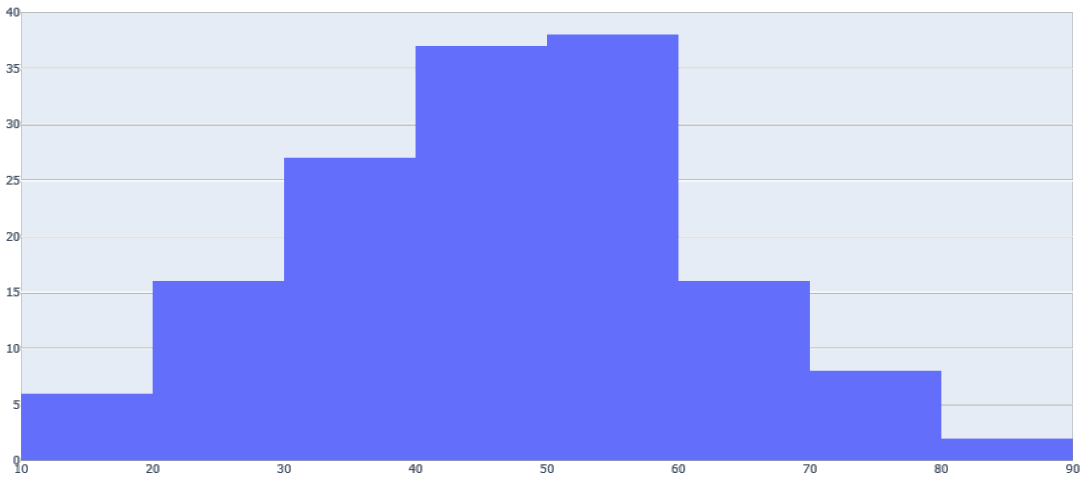
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Vector layer histogram

Generates a histogram with the values of the attribute of a vector layer.

The attribute to use for computing the histogram must be numeric.



24.38 Fig. : Example of vector layer histogram

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vector: any]	Įvesties vektorinis sluoksnis
Atributas	`FIELD`	[lentelės laukas: skaičius]	Reikšmė, kuri naudojama braižymui (Y ašis)
number of bins	`BINS`	[skaičius: sveikas skaičius] Numatytas: 10	The number of bins to use in the histogram (X axis). Minimum 2.
Histogram	`OUTPUT`	[html] Numatytas: `[Įrašyti į laikiną failą]`	Įrašykite brėžinio HTML failą. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Histogram	`OUTPUT`	[html]	HTML failas su brėžiniu. Galima rasti *Apdorojimas ► Rezultatų žiūryklė*.

Pythono kodas

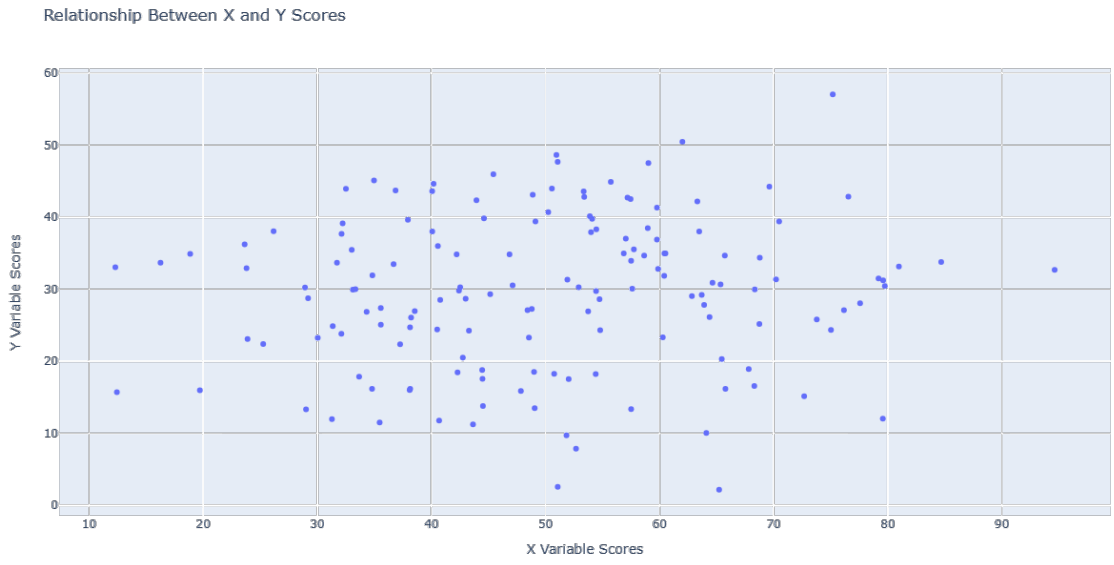
Algorithm ID: `qgis:vectorlayerhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Vector layer scatterplot

Creates a simple `X` - `Y` scatter plot for a vector layer.



24.39 Fig. : Example of vector layer scatterplot

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
X attribute	`XFIELD`	[lentelės laukas: skaičius]	Field to use for the X axis
Y attribute	`YFIELD`	[lentelės laukas: skaičius]	Field to use for the Y axis
Hover text Optional	`HOVERTEXT`	[expression] Default: „“	Text to be shown when hovering with the mouse over data points. It can be picked from a field or based on an expression.
Title Optional	`TITLE`	[string] Default: „“	Title of the plot
X-axis Title Optional	`XAXIS_TITLE`	[string] Default: „“	If empty, the field name of the x attribute is used. With a single space, the axis title is not shown.
Y-axis Title Optional	`YAXIS_TITLE`	[string] Default: „“	If empty, the field name of the y attribute is used. With a single space, the axis title is not shown.
Use logarithmic scale for x-axis	`XAXIS_LOG`	[boolean] Default: False	When enabled, uses logarithmic scale for the x-axis
Use logarithmic scale for y-axis	`YAXIS_LOG`	[boolean] Default: False	When enabled, uses logarithmic scale for the y-axis
Scatterplot	`OUTPUT`	[html] Numatytas: `[Įrašyti į laikiną failą]`	Įrašykite brėžinio HTML failą. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Scatterplot	`OUTPUT`	[html]	HTML failas su brėžiniu. Galima rasti *Apdorojimas* ► *Rezultatų žiūryklė*.

Pythono kodas

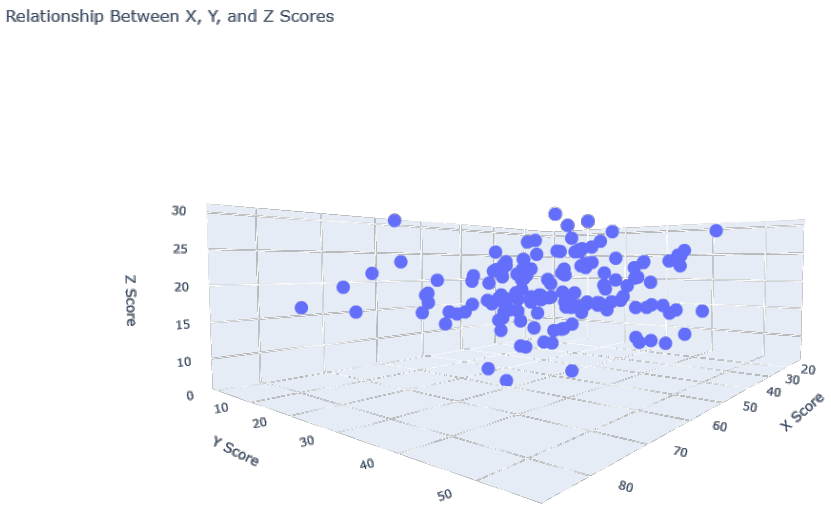
Algorithm ID: `qgis:vectorlayerscatterplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Vector layer scatterplot 3D

Creates a 3D scatter plot for a vector layer.



24.40 Fig. : Example of 3D scatter plot

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
X attribute	`XFIELD`	[lentelės laukas: skaičius]	Field to use for the X axis
Y attribute	`YFIELD`	[lentelės laukas: skaičius]	Field to use for the Y axis
Z attribute	`ZFIELD`	[lentelės laukas: skaičius]	Field to use for the Z axis
Title Optional	`TITLE`	[string] Default: „“	Title of the plot
X-axis Title Optional	`XAXIS_TITLE`	[string] Default: „“	If empty, the field name of the X attribute is used.
Y-axis Title Optional	`YAXIS_TITLE`	[string] Default: „“	If empty, the field name of the Y attribute is used.
Z-axis Title Optional	`ZAXIS_TITLE`	[string] Default: „“	If empty, the field name of the Z attribute is used.
Scatterplot 3D	`OUTPUT`	[html] Numatytas: `[Įrašyti į laikiną failą]`	Įrašykite brėžinio HTML failą. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Scatterplot 3D	`OUTPUT`	[html]	HTML failas su brėžiniu. Galima rasti *Apdorojimas* ► *Rezultatų žiūryklė*.

Pythono kodas

Algorithm ID: `qgis:scatter3dplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.15 Taškų masyvo keitimas

Dėmesio

Running these algorithms requires QGIS installed with PDAL >= 2.5.0 (see *Help* ► *About* menu).

Keisti formatą

Keičia taškų debesį į kitą failo formatą, pavyzdžiui sukuria suspaustą `.LAZ`.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[taškų masyvas]	Įvesties taškų masyvo sluoksnis, kurį reikia keisti
Pakeistas	`OUTPUT`	[taškų masyvas] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite taškų masyvo failą, kurį naudoti kaip išvestį. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Pakeistas	`OUTPUT`	[taškų masyvas]	Išvesties taškų masyvo sluoksnis pakeistu failo formatu. Šiuo metu palaikomi formatai yra `.LAS`, `.LAZ`, `.COPC.LAZ` ir `.VPC`.

Pythono kodas

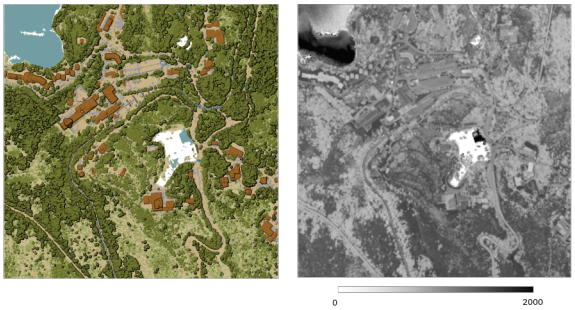
Algoritmo ID: `pdal:convertformat`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Eksportuoti į rastrą

Eksportuoja taškų masyvo duomenis į 2D rastro gardelę su nurodytos rezoliucijos celės dydžiu, įrašant reikšmes iš nurodyto atributo.



24.41 Fig. : Rastro išvestis naudojant taškų intensyvumo atributą

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[taškų masyvas]	Įvesties taškų masyvo sluoksnis eksportavimui
Atributas	`ATTRIBUTE`	[laukas] [sąrašas]	Taškų masyvo sluoksnio laukas, iš kurio reikia ištraukti reikšmes
Tankio rastro rezoliucija	`RESOLUTION`	[skaičius: slankaus kablelio] Numatytas: 1.0	Išvesties rastro celės dydis

continues on next page

Table 24.258 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Paralelinio vykdymo kaladėlės dydis	`TILE_SIZE`	[skaičius: sveikas skaičius] Numatytas: 1000	
Eksportuota	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite rastro failą, į kurį eksportuoti duomenis. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Filtro išraiška Pasirinktinis	`FILTER_EXPRESS`	[išraiška]	*PDAL išraiška* parenkant taškų masyvo duomenų geoobjektų poaibį
Apkarpymo apimtis Pasirinktinis	`FILTER_EXTENT`	[apimtis]	Žemėlapio apimtis, parenkant taškų masyvo duomenų geoobjektų poaibį Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Paralelinio vykdymo kaladėlių pradžia X Pasirinktinis	`ORIGIN_X`	[skaičius: slankaus kablelio]	
Paralelinio vykdymo kaladėlių pradžia Y Pasirinktinis	`ORIGIN_Y`	[skaičius: slankaus kablelio]	

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Eksportuota	`OUTPUT`	[rastras]	Išvesties rastro sluoksnio geoobjektai eksportuoti iš taškų masyvo sluoksnio. Šiuo metu palaikomas formatas yra `.TIF`.

Pythono kodas

Algoritmo ID: `pdal:exportraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

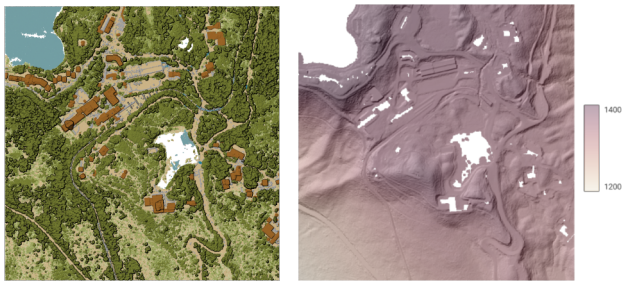
algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Eksportuoti rastrą (naudojant trianguliaciją)

Eksportuoja taškų masyvo duomenis į 2D rastrą naudojant taškų trianguliaciją ir tada interpoliuojant celių reikšmes iš trikampių.

Pastaba

Šis algoritmas gali būti lėtas, jei dirbate su dideliu duomenų rinkiniu. Jei jūsų taškų masyvas yra tankus, jūs galite eksportuoti žemės taškus kaip rastrą, naudodami algoritmą *Eksportuoti į rastrą*.



24.42 Fig. : Paviršiaus rastro išvestis, sukurta naudojant taškų masyvo trianguliaciją

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[taškų masyvas]	Įvesties taškų masyvo sluoksnis eksportavimui

continues on next page

Table 24.261 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Tankio rastro rezoliucija	`RESOLUTION`	[skaičius: slankaus kablelio] Numatytas: 1.0	Išvesties rastro celės dydis
Paralelinio vykdymo kaladėlės dydis	`TILE_SIZE`	[skaičius: sveikas skaičius] Numatytas: 1000	
Eksportuota	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite rastro failą, į kurį eksportuoti duomenis. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Filtro išraiška Pasirinktinis	`FILTER_EXPRESS`	[išraiška]	*PDAL išraiška* parenkant taškų masyvo duomenų geoobjektų poaibį
Apkarpymo apimtis Pasirinktinis	`FILTER_EXTENT`	[apimtis]	Žemėlapio apimtis, parenkant taškų masyvo duomenų geoobjektų poaibį Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Paralelinio vykdymo kaladėlių pradžia X Pasirinktinis	`ORIGIN_X`	[skaičius: slankaus kablelio]	
Paralelinio vykdymo kaladėlių pradžia Y Pasirinktinis	`ORIGIN_Y`	[skaičius: slankaus kablelio]	

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Eksportuotas (naudojant trianguliaciją)	`OUTPUT`	[rastras]	Išvesties rastro sluoksnio geoobjektai eksportuoti iš taškų masyvo sluoksnio. Šiuo metu palaikomas formatas yra `.TIF`.

Pythono kodas

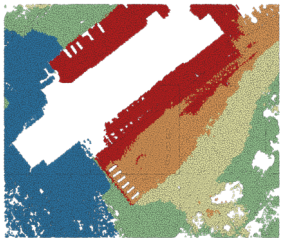
Algoritmo ID: `pdal:exportrastertin`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export to vector

Exports point cloud data to a vector layer with 3D points (a GeoPackage), optionally with extra attributes.



24.43 Fig. : Exporting point cloud (ground points) to a vector layer styled based on the elevation

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[taškų masyvas]	Įvesties taškų masyvo sluoksnis eksportavimui
Atributas Pasirinktinis	`ATTRIBUTE`	[field] [list]	One or more fields of the point cloud layer to export with the points.
Eksportuota	`OUTPUT`	[vector] Numatytas: `[Įrašyti į laikiną failą]`	Specify the vector file to export the data to. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Filtro išraiška Pasirinktinis	`FILTER_EXPRESS`	[išraiška]	*PDAL išraiška* parenkant taškų masyvo duomenų geoobjektų poaibį
Apkarpymo apimtis Pasirinktinis	`FILTER_EXTENT`	[apimtis]	Žemėlapio apimtis, parenkant taškų masyvo duomenų geoobjektų poaibį Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Eksportuota	`OUTPUT`	[vector]	Output vector layer features of the point cloud layer are exported to. Currently supported format is `.GPKG`.

Pythono kodas

Algorithm ID: `pdal:exportvector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.16 Taškų debesies duomenų valdymas

Dėmesio

Running these algorithms requires QGIS installed with PDAL >= 2.5.0 (see *Help* ► *About* menu).

Priskirti projekciją

Priskiria taškų masyvo sluoksniui koordinačių atskaitos sistemą, jei jos nėra ar ji neteisinga. Sukuriamas naujas sluoksnis.

Taip pat žiūrėkite

Perprojektuoti

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[taškų masyvas]	Įvesties taškų masyvo sluoksnis, kuriam reikia priskirti CRS
Norima CRS	`CRS`	[crs]	Sluoksniui priskiriama CRS
Išvesties sluoksnis	`OUTPUT`	[taškų masyvas] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite taškų masyvo failą, kurį naudoti kaip išvestį. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Pakeistas	`OUTPUT`	[taškų masyvas]	Išvesties taškų masyvo sluoksnis su nauja CRS. Šiuo metu palaikomi formatai yra: `.LAS`, `.LAZ`, `.COPC.LAZ` ir `.VPC`.

Pythono kodas

Algoritmo ID: `pdal:assignprojection`

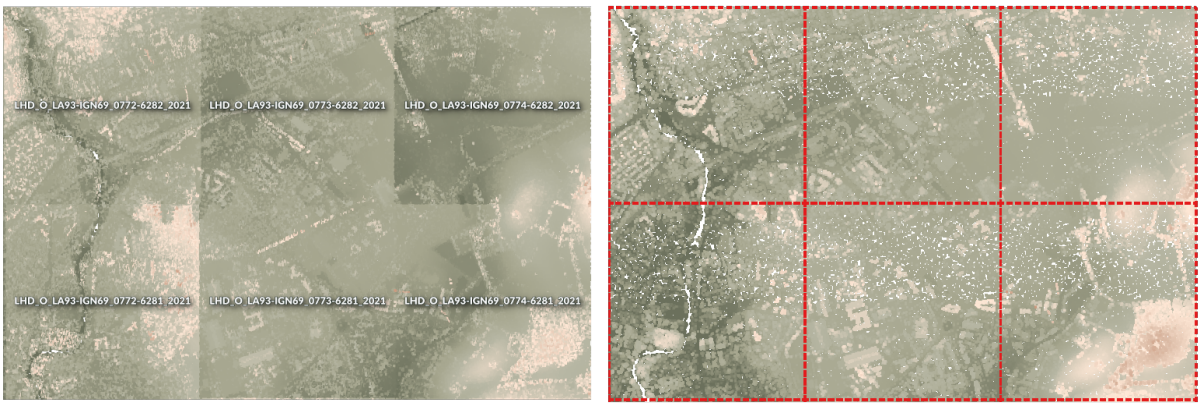
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Kurti virtualų taškų masyvą (VPC)

Kuria *virtualų taškų masyvą (VPC)* iš įvesties taškų masyvo duomenų.

Jei paliksite neprivalomus parametrus neįjungtus, VPC failas bus sukurtas labai greitai, nes algoritmas tik perskaitys įvesties failų metaduomenis. Įjungus bet kurį iš pasirinktinių parametrų, algoritmas perskaitys visus taškus, o tai gali užimti nemažai laiko.



24.44 Fig. : Virtualaus taškų masyvo su apžvalga kūrimas iš kelių taškų masyvų failų

Taip pat žiūrėkite

Sulieti

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`LAYERS`	[taškų masyvas] [sąrašas]	Įvesties taškų masyvo sluoksniai, kuriuos reikia sujungti į virtualų taškų masyvo sluoksnį
Skaičiuoti ribos poligoną	`BOUNDARY`	[boolean] Numatytas: ne	Nustatykite True, kad būtų rodoma tiksli duomenų riba (o ne tik stačiakampė apimtis)
Skaičiuoti statistiką	`STATISTICS`	[boolean] Numatytas: ne	Nustatykite True, kad suprastumėte įvairių atributų reikšmių diapazonus
Kurti apžvalgos taškų masyvą	`OVERVIEW`	[boolean] Numatytas: ne	Kuria vieną „praretintą“ taškų masyvą visiems įvesties duomenims (naudojant tik kas 1000-ąjį pradinių duomenų tašką). Apžvalgos taškų masyvas bus sukurtas greta VPC failo - pavyzdžiui, jei failas yra `manoduomenys.vpc`, tai apžvalgos taškų masyvo pavadinimas bus `manoduomenys-overview.copc.laz`.
Virtualus taškų masyvas	`OUTPUT`	[taškų masyvas] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite taškų masyvo failą, į kurį sudėti duomenis. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Virtualus taškų masyvas	`OUTPUT`	[rastras]	Išvesties taškų masyvo sluoksnis, apjungiantis visus įvesties duomenis kaip virtualų failą.

Pythono kodas

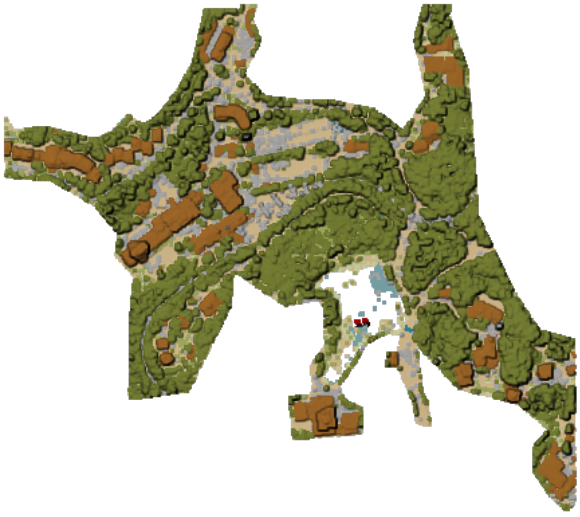
Algoritmo ID: `pdal:virtualpointcloud`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Apkirpti

Apkerpa taškų masyvo sluoksnį pagal poligono sluoksnį taip, kad gautame taškų masyve yra tik taškai, patenkantys į poligonus.



24.45 Fig. : Įvesties taškų masyvo apkarpymas poligonų padengimu

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[taškų masyvas]	Apkarpomas įvesties taškų masyvo sluoksnis
Apkarpymo poligonai	`OVERLAY`	[vektorius: poligonas]	Poligonų vektorinis sluoksnis, kuris naudojamas kaip apkarpomų taškų padengimas
Apkirpta	`OUTPUT`	[taškų masyvas] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite taškų masyvo failą, į kurį reikia eksportuoti apkarpytus taškus. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Filtro išraiška Pasirinktinis	`FILTER_EXPRESS`	[išraiška]	*PDAL išraiška* parenkant taškų masyvo duomenų geoobjektų poaibį
Apkarpymo apimtis Pasirinktinis	`FILTER_EXTENT`	[apimtis]	Žemėlapio apimtis, parenkant taškų masyvo duomenų geoobjektų poaibį Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Apkirpta	`OUTPUT`	[taškų masyvas]	išvesties taškų masyvas, kurio geoobjektai yra taškai yra dengiami poligonų sluoksnio.

Pythono kodas

Algoritmo ID: `pdal:clip`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Kurti COPC

Sukuria indeksus visiems įvesties taškų masyvų failams paketiniame režime.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`LAYERS`	[taškų masyvas] [sąrašas]	Įvesties taškų masyvų sluoksniai, kuriems reikia sukurti indeksus
Išvesties aplankas Pasirinktinis	`OUTPUT`	[aplankas] Numatytas: `[Praleisti išvestį]`	Nurodykite aplanką, kuriame kurti naujus failus. *Vienas iš*: • Praleisti išvestį • Įrašyti į laikiną aplanką • Įrašyti į aplanką

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties aplankas	`OUTPUT`	[aplankas]	Išvesties aplankas, kuriame bus taškų masyvų sluoksniai su kartu einančiais COPC indeksų failais.

Pythono kodas

Algoritmo ID: `pdal:createcopc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Informacija

Grąžina bazinius įvesties taškų masyvo failo metaduomenis.

Išvesties metaduomenų pavyzdys:

```
LAS           1.4
point format  6
count         56736130
scale         0.001 0.001 0.001
offset        431749.999 5440919.999 968.898
extent        431250 5440420 424.266
              432249.999 5441419.999 1513.531
crs           ETRS89 / UTM zone 34N (N-E) (EPSG:3046)  (vertical CRS missing!)
units         horizontal=metre  vertical=unknown

Attributes:
- X floating 8
- Y floating 8
- Z floating 8
- Intensity unsigned 2
- ReturnNumber unsigned 1
- NumberOfReturns unsigned 1
- ScanDirectionFlag unsigned 1
- EdgeOfFlightLine unsigned 1
- Classification unsigned 1
- ScanAngleRank floating 4
- UserData unsigned 1
- PointSourceId unsigned 2
- GpsTime floating 8
- ScanChannel unsigned 1
- ClassFlags unsigned 1
```

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[taškų masyvas]	Įvesties taškų masyvo sluoksnis, iš kurio reikia ištraukti metaduomenų informaciją
Sluoksnio informacija	`OUTPUT`	[failas] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite failą, kuriame saugoti metaduomenų informaciją. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sluoksnio informacija	`OUTPUT`	[html]	`HTML` failas, kuriame laikyti metaduomenų informaciją.

Pythono kodas

Algoritmo ID: `pdal:info`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Sulieti

Sulieja kelis taškų masyvų failus į vieną.

Taip pat žiūrėkite

Kurti virtualų taškų masyvą (VPC)

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`LAYERS`	[taškų masyvas] [sąrašas]	Įvesties taškų masyvų sluoksniai, kuriuos reikia sulieti į vieną.
Sulietas	`OUTPUT`	[taškų masyvas] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite sulieto išvesties taškų masyvo failą. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Filtro išraiška Pasirinktinis	`FILTER_EXPRESS`	[išraiška]	*PDAL išraiška* parenkant taškų masyvo duomenų geoobjektų poaibį

continues on next page

Table 24.279 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Apkarpymo apimtis Pasirinktinis	`FILTER_EXTENT`	[apimtis]	Žemėlapio apimtis, parenkant taškų masyvo duomenų geoobjektų poaibį Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sulietas	`OUTPUT`	[taškų masyvas]	Išvesties taškų masyvo sluoksnis, suliejantis visus įvesties failus.

Pythono kodas

Algoritmo ID: `pdal:merge`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Perprojektuoti

Perprojektuoja taškų masyvą į kitą koordinačių atskaitos sistemą (CRS).

Taip pat žiūrėkite

Priskirti projekciją

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[taškų masyvas]	Įvesties taškų masyvo sluoksnis, kur reikia perprojektuoti į kitą CRS
Paskirties CRS	`CRS`	[crs]	Sluoksniui priskiriama CRS
Perprojektuota	`OUTPUT`	[taškų masyvas] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite perprojektuojamą taškų masyvo failą. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Koordinačių veiksmas Pasirinktinis	`OPERATION`	[datum]	*Duomenų transformacija*, kurią reikia naudoti perprojektuojant duomenis tarp pradinės ir paskirties sistemų.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
PERPROJEKTUO	`OUTPUT`	[taškų masyvas]	Išvesties taškų masyvo sluoksnis su paskirties CRS

Pythono kodas

Algoritmo ID: `pdal:reproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Ploninti (pagal mėginio spindulį)

Sukuria paplonintą taškų masyvo versiją imant mėginius pagal atstumo tašką (sumažina kiekį taškų, esančių konkrečiame spindulyje).

Taip pat žiūrėkite

Ploninti (praleidžiant taškus)

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[taškų masyvas]	Įvesties taškų masyvo sluoksnis, kurio plonintą versiją reikia sukurti
Mėginių spindulys (žemėlapio vienetais)	`SAMPLING_RADIU`	[skaičius: slankaus kablelio] Numatytas: 1.0	Atstumas, kuriame taškai surenkami į vieną unikalų tašką
Paplonintas (pagal spindulį)	`OUTPUT`	[taškų masyvas] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite išvesties taškų masyvą su sumažintu taškų skaičiumi. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Filtro išraiška Pasirinktinis	`FILTER_EXPRESS`	[išraiška]	*PDAL išraiška* parenkant taškų masyvo duomenų geoobjektų poaibį
Apkarpymo apimtis Pasirinktinis	`FILTER_EXTENT`	[apimtis]	Žemėlapio apimtis, parenkant taškų masyvo duomenų geoobjektų poaibį Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Paplonintas (pagal spindulį)	`OUTPUT`	[taškų masyvas]	Išvesties taškų masyvo sluoksnis su sumažintu taškų skaičiumi.

Pythono kodas

Algoritmo ID: `pdal:thinbyradius`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Ploninti (praleidžiant taškus)

Sukuria paplonintą taškų masyvo versiją paliekant tik kas N-tąjį tašką (sumažina taškų skaičių praleidžiant greta esančius taškus).

Taip pat žiūrėkite

Ploninti (pagal mėginio spindulį)

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[taškų masyvas]	Įvesties taškų masyvo sluoksnis, kurio plonintą versiją reikia sukurti
Praleidžiamų taškų skaičius	`POINTS_NUMBER`	[skaičius: sveikas skaičius] Numatytas: 1	Palikti tik kas N-tą įvesties sluoksnio tašką
Paplonintas (sunaikinimu)	`OUTPUT`	[taškų masyvas] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite išvesties taškų masyvą su sumažintu taškų skaičiumi. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Filtro išraiška Pasirinktinis	`FILTER_EXPRESS`	[išraiška]	*PDAL išraiška* parenkant taškų masyvo duomenų geoobjektų poaibį

continues on next page

Table 24.288 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Apkarpymo apimtis Pasirinktinis	`FILTER_EXTENT`	[apimtis]	Žemėlapio apimtis, parenkant taškų masyvo duomenų geoobjektų poaibį Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Paplonintas (sunaikinimu)	`OUTPUT`	[taškų masyvas]	Išvesties taškų masyvo sluoksnis su sumažintu taškų skaičiumi.

Pythono kodas

Algoritmo ID: `pdal:thinbydecimate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Kaladėlės

Kuria įvesties taškų masyvo failų kaladėles, rekomenduojama geriausiai greitaveikai (rodant ar analizuojant) dirbant su tokiais duomenų rinkiniais QGIS.

Taip pat žiūrėkite

Kurti virtualų taškų masyvą (VPC), *Kurti COPC*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`LAYERS`	[taškų masyvas] [sąrašas]	Įvesties taškų masyvo sluoksniai, iš kurių kurti kaladėles
Kaladėlės ilgis	`LENGTH`	[skaičius: slankaus kablelio] Numatytas: 1000.0	Kiekvienos sugeneruotos kaladėlės kraštinės dydis
Išvesties aplankas	`OUTPUT`	[aplankas] Numatytas: `[Įrašyti į laikiną aplanką]`	Nurodykite aplanką, kuriame rašyti sukurtas kaladėles. *Vienas iš*: • Įrašyti į laikiną aplanką • Įrašyti į aplanką

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Priskirti CRS Pasirinktinis	`CRS`	[crs]	Sluoksniui priskiriama CRS

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties aplankas	`OUTPUT`	[aplankas]	Išvesties aplankas, kuriame yra iš įvesties failų sugeneruotos kaladėlės.

Pythono kodas

Algoritmo ID: `pdal:tile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

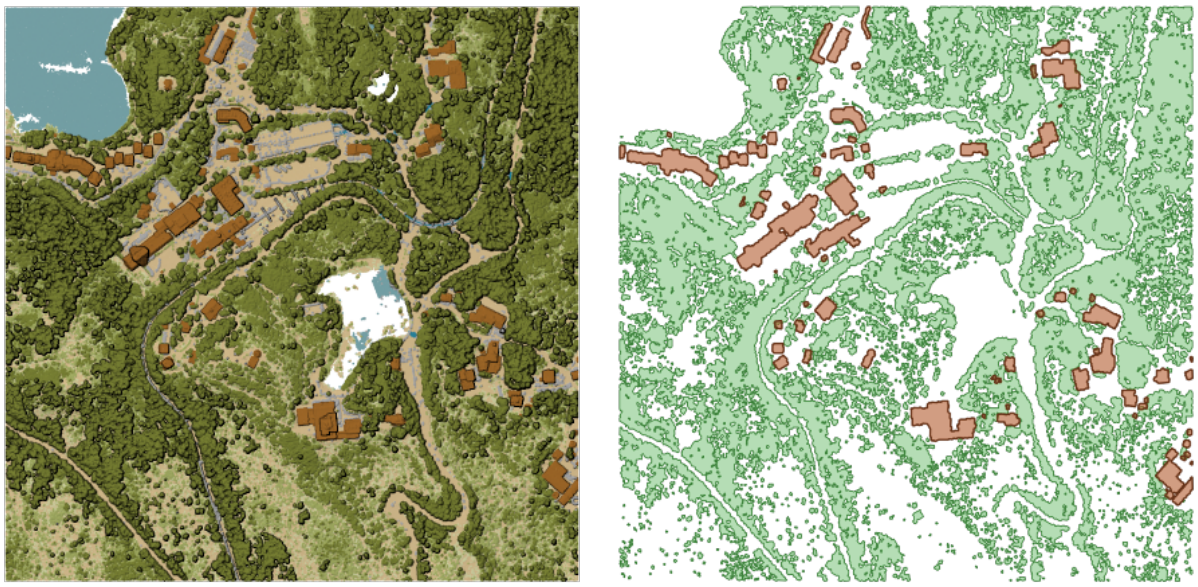
24.1.17 Taškų masyvo ištraukimas

Dėmesio

Running these algorithms requires QGIS installed with PDAL >= 2.5.0 (see *Help* ► *About* menu).

Ribos

Eksportuoja poligono failą su taškų masyvo sluoksnio ribomis. Jame gali būti skylės ir tai gali būti kelių dalių poligonas.



24.46 Fig. : Aukštos augmenijos ir pastatų poligonų ištraukimas iš įvesties taškų masyvo sluoksnio

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[taškų masyvas]	Įvesties taškų masyvo sluoksnis ribų skaičiavimui
Ribos skaičiavimui naudojamų celių rezoliucija Pasirinktinis	`RESOLUTION`	[skaičius: slankaus kablelio]	Ribos skaičiavimui naudojamų celių rezoliucija
Minimalus skaičius taškų celėje, kad ji būtų laikoma užimta Pasirinktinis	`THRESHOLD`	[skaičius: sveikas skaičius]	Minimalus skaičius taškų celėje, kad ji būtų laikoma užimta

continues on next page

Table 24.293 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Riba	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite taškų masyvo failą, kurį naudoti kaip išvestį. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Filtro išraiška Pasirinktinis	`FILTER_EXPRESS`	[išraiška]	*PDAL išraiška* parenkant taškų masyvo duomenų geoobjektų poaibį
Apkarpymo apimtis Pasirinktinis	`FILTER_EXTENT`	[apimtis]	Žemėlapio apimtis, parenkant taškų masyvo duomenų geoobjektų poaibį Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Riba	`OUTPUT`	[vektorius: poligonas]	Išvesties poligonų vektorinis sluoksnis su taškų masyvo ribomis. Šiuo metu palaikomas formatas yra `.GPKG`.

Pythono kodas

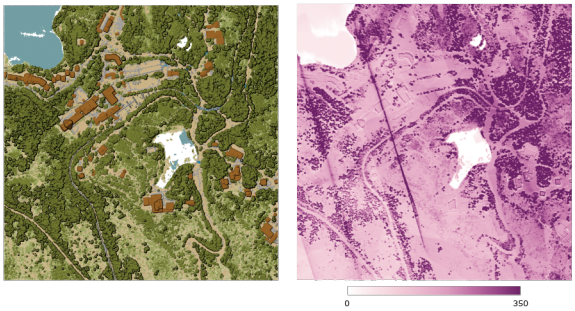
Algoritmo ID: `pdal:boundary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Tankumas

Eksportuoja rastro failą pagal kiekvienoje rastro celėje esančių taškų skaičių - naudinga taškų masyvų duomenų aibių kokybės tikrinimui.



24.47 Fig. : Taškų tankumas (taškų skaičius 2x2 m) kaip rastras

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[taškų masyvas]	Įvesties taškų masyvo sluoksnis eksportavimui
Tankio rastro rezoliucija	`RESOLUTION`	[skaičius: slankaus kablelio] Numatytas: 1.0	Išvesties rastro celės dydis
Paralelinio vykdymo kaladėlės dydis	`TILE_SIZE`	[skaičius: sveikas skaičius] Numatytas: 1000	Kaladėlių dydis, dalinant duomenis paraleliniam skaičiavimui
Tankumas	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite rastro failą, į kurį eksportuoti duomenis. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Filtro išraiška Pasirinktinis	`FILTER_EXPRESS`	[išraiška]	*PDAL išraiška* parenkant taškų masyvo duomenų geoobjektų poaibį
Apkarpymo apimtis Pasirinktinis	`FILTER_EXTENT`	[apimtis]	Žemėlapio apimtis, parenkant taškų masyvo duomenų geoobjektų poaibį Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Paralelinio vykdymo kaladėlių pradžia **X** Pasirinktinis	`ORIGIN_X`	[skaičius: slankaus kablelio]	
Paralelinio vykdymo kaladėlių pradžia **Y** Pasirinktinis	`ORIGIN_Y`	[skaičius: slankaus kablelio]	

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Tankumas	`OUTPUT`	[rastras]	Išvesties rastro sluoksnis su kiekvienoje celėje esančių taškų skaičiumi. Šiuo metu palaikomas formatas yra `.TIF`.

Pythono kodas

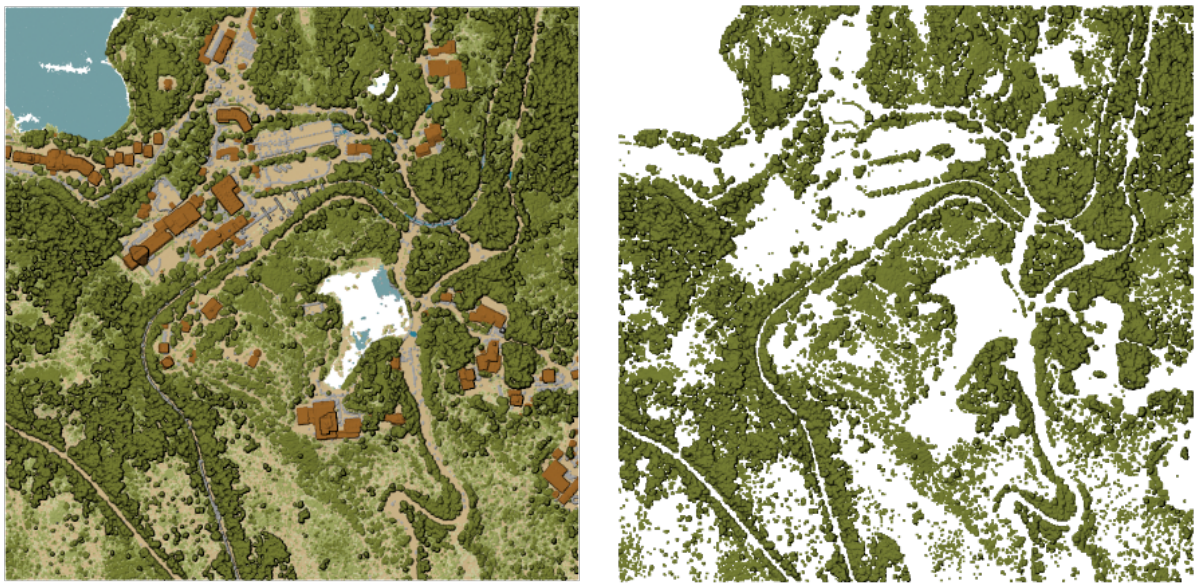
Algoritmo ID: `pdal:density`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Filter

Extracts point from the input point cloud which match PDAL expression and/or are inside of a cropping rectangle.



24.48 Fig. : Filtering of high vegetation class from an input point cloud layer

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[taškų masyvas]	Įvesties taškų masyvo sluoksnis eksportavimui
Filtro išraiška Pasirinktinis	`FILTER_EXPRESS`	[išraiška]	*PDAL išraiška* parenkant taškų masyvo duomenų geoobjektų poaibį

continues on next page

Table 24.299 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Apkarpymo apimtis Pasirinktinis	`FILTER_EXTENT`	[apimtis]	Žemėlapio apimtis, parenkant taškų masyvo duomenų geoobjektų poaibį Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Filtered	`OUTPUT`	[taškų masyvas] Numatytas: `[Įrašyti į laikiną failą]`	Specify the point cloud file to export the data to. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Filtered	`OUTPUT`	[taškų masyvas]	Output point cloud layer with the filtered features of the input point cloud layer. Currently supported formats are `.LAS`, `.LAZ`, `.COPC.LAZ` and `.VPC`.

Pythono kodas

Algorithm ID: `pdal:filter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.18 Rastro analizė

Celių krūvos procentinis rangas pagal reikšmę

Skaičiuoja rastrų krūvos celių-lygmens procentinį rangą paremta viena įvesties reikšme ir įrašo jį į išvesties rastrą.

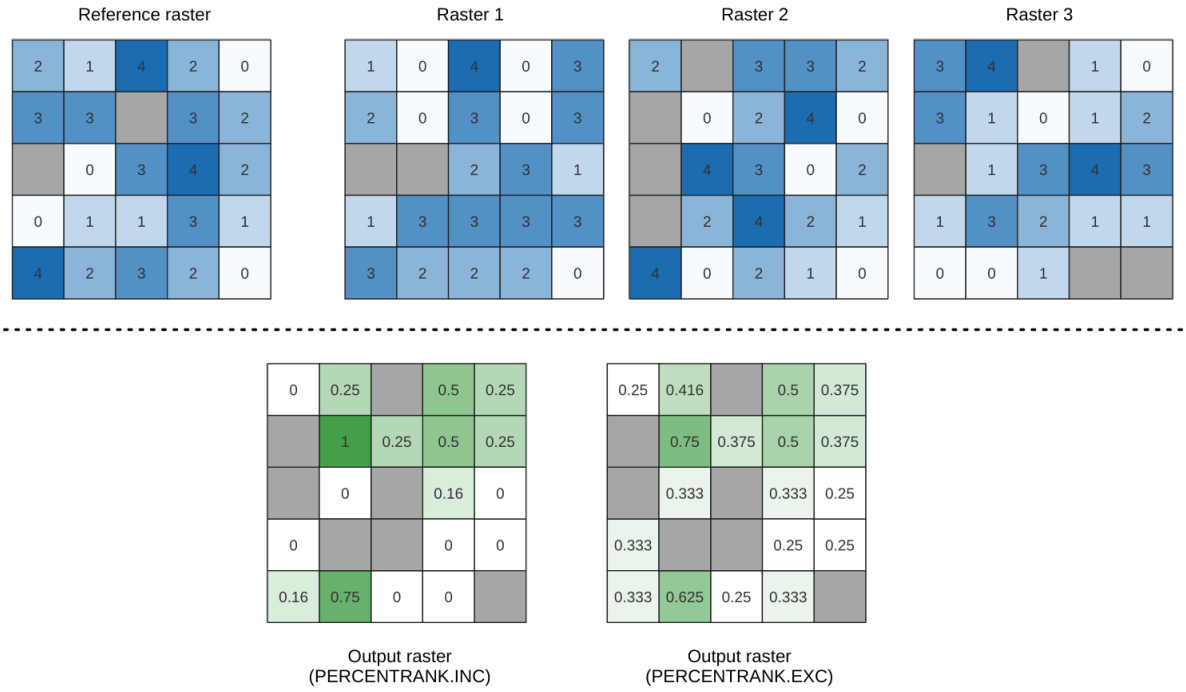
Kiekvienoje celės vietoje, nurodyta reikšmė lyginama su atitinkamomis krūvos įvesties sluoksnių reikšmėmis visuose perdengtose ir rikiuotose celėse. Reikšmėms už krūvos reikšmių išsidėstymo, algoritmas grąžina NėraDuomenų, nes reikšmės negalima priskirti rangui.

Yra du procentilių skaičiavimo metodai:

- Tiesinė interpoliacija imtinai (PERCENTRANK.INC)
- Tiesinė interpoliacija išimtinai (PERCENTRANK.EXC)

Tiesinės interpoliacijos metodai skirtingoms reikšmėms grąžina unikalų procentinį rangą. Abu interpoliacijos metodai veikia pagal atitinkamus LibreOffice ar Microsoft Excel įgyvendinimus.

Išvesties rastro apimtis ir rezoliucija nustatoma pagal atskaitos rastrą. Įvesties rastro sluoksniai, kurie neatitinka atskaitos rastro sluoksnio celės dydžio, bus perrinkti naudojant artimiausio kaimyno metodą. Įvesties sluoksnių reikšmės NėraDuomenų reikš NėraDuomenų išvestį, jei nenurodytas parametras „Ignoruoti NėraDuomenų reikšmes“. Išvesties rastro duomenų tipas visada bus `Float32`.



24.49 Fig. : Procentinio rango reikšmė = 1. `NėraDuomenų` celės (pilkos) ignoruojamos.

Taip pat žiūrėkite

Celių krūvos procentilis, *Cell stack percentrank from raster layer*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`INPUT`	[rastras] [sąrašas]	Vertinami rastro sluoksniai. Jei rastro duomenų krūvoje naudojami kelių juostų rastrai, algoritmas analizę visada vykdys pirmoje rastro juostoje
Metodas	`METHOD`	[sąrašas] Numatytas: 0	Percentilės skaičiavimo metodas: • 0 — Tiesinė interpoliacija imtinai (PERCENTRANK.INC) • 1 — Tiesinė interpoliacija išimtinai (PERCENTRANK.EXC)
Reikšmė	`VALUE`	[skaičius: slankaus kablelio] Numatytas: 10.0	Reikšmė, kurios rangą reikia rasti, atitinkamai reikšmėms krūvoje visų perdengtų ir rikiuotų įvesties rastro celių reikšmių
Ignoruoti NėraDuomenų reikšmes	`IGNORE_NODATA`	[boolean] Numatytas: Taip	Neįjungus šios parinkties, visos įvesties sluoksnių NėraDuomenų celės reikš NėraDuomenų celes išvesties rastre
Atskaitos sluoksnis	`REFERENCE_LAYE`	[rastras]	Išvesties sluoksnio kūrimo atskaitos sluoksnis (apimtis, CRS, taškų dimensijos)
Išvesties sluoksnis	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties NėraDuomenų reikšmė	`OUTPUT_NODATA_`	[skaičius: slankaus kablelio] Numatyta: -9999.0	NėraDuomenų reikšmė, kurią reikia naudoti išvesties sluoksnyje
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	`OUTPUT`	[rastras]	Išvesties rastras su rezultatu
CRS identifikatorius	`CRS_AUTHID`	[tekstas]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	`EXTENT`	[tekstas]	Išvesties rastro erdvinė apimtis
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius
Aukštis taškais	`HEIGHT_IN_PIXE`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	`TOTAL_PIXEL_CO`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius

Pythono kodas

Algoritmo ID: `native:cellstackpercentrankfromvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Celių krūvos procentilis

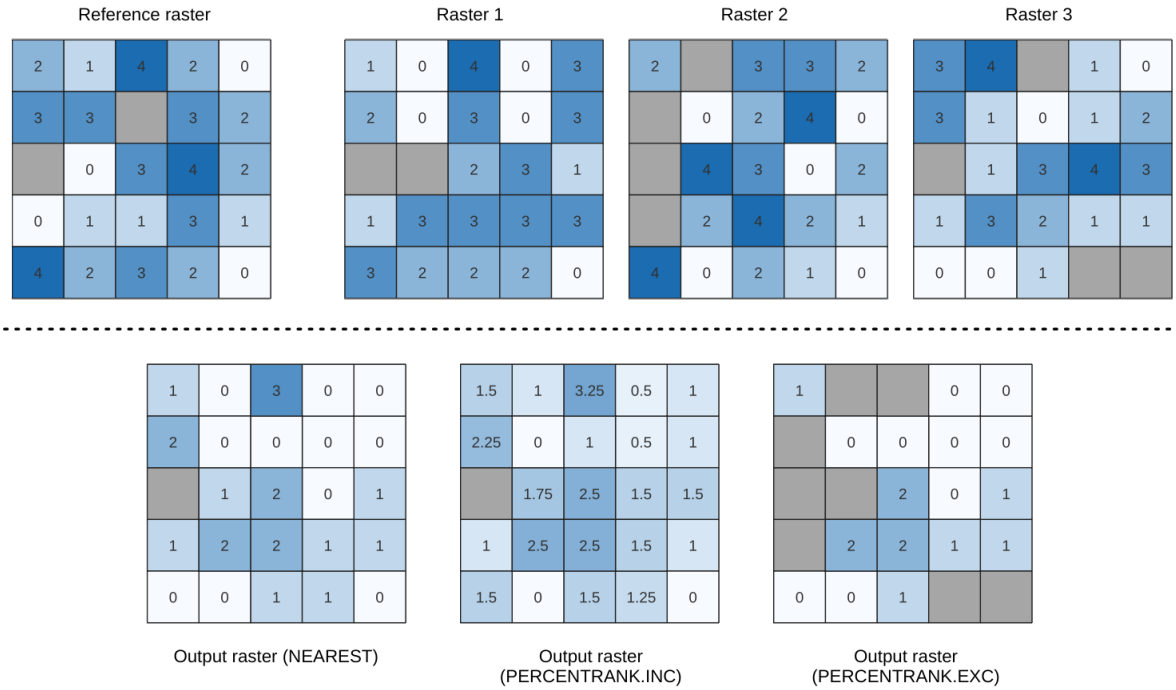
Skaičiuoja celių krūvos procentilio reikšmę rastrų krūvai ir įrašo rezultatus į išvesties rastrą. Grąžinanas procentilis skaičiuojamas pagal procentilio įvesties reikšmę (diapazonas nuo 0 iki 1). Kiekvienoje celės pozicijoje nurodytas procentilis gaunamas naudojamas atitinkamas reikšmes iš krūvos visų perdengtų ir surikiuotų įvesties rastro reikšmių.

Yra trys procentilio skaičiavimo būdai:

- Artimiausias rangas: grąžina reikšmę, kuri yra artimiausia nurodytam procentiliui
- Tiesinė interpoliacija imtinai (PERCENTRANK.INC)
- Tiesinė interpoliacija išimtinai (PERCENTRANK.EXC)

Tiesinės interpoliacijos metodai skirtingiems procentiliams grąžina unikalią reikšmę. Abu interpoliacijos metodai veikia pagal atitinkamus LibreOffice ar Microsoft Excel įgyvendinimus.

Išvesties rastro apimtis ir rezoliucija nustatoma pagal atskaitos rastrą. Įvesties rastro sluoksniai, kurie neatitinka atskaitos rastro sluoksnio celės dydžio, bus perrinkti naudojant artimiausio kaimyno metodą. Įvesties sluoksnių reikšmės NėraDuomenų reikš NėraDuomenų išvestį, jei nenurodytas parametras „Ignoruoti NėraDuomenų reikšmes“. Išvesties rastro duomenų tipas visada bus `Float32`.



24.50 Fig. : Procentilis = 0.25. `NėraDuomenų` celės (pilkos) ignoruojamos.

Taip pat žiūrėkite

Celių krūvos procentilis, *Cell stack percentrank from raster layer*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`INPUT`	[rastras] [sąrašas]	Vertinami rastro sluoksniai. Jei rastro duomenų krūvoje naudojami kelių juostų rastrai, algoritmas analizę visada vykdys pirmoje rastro juostoje
Metodas	`METHOD`	[sąrašas] Numatytas: 0	Percentilės skaičiavimo metodas: • 0 — Artimiausias rangas: grąžina reikšmę, kuri yra artimiausia nurodytam procentiliui • 1 — Tiesinė interpoliacija imtinai (PERCENTILE.INC) • 2 — Tiesinė interpoliacija išimtinai (PERCENTILE.EXC)
Procentilis	`VALUE`	[skaičius: slankaus kablelio] Numatyta: 0.25	Reikšmė, kurios rangą reikia rasti, atitinkamai reikšmėms krūvoje visų perdengtų ir rikiuotų įvesties rastro celių reikšmių. Tarp 0 ir 1.

continues on next page

Table 24.304 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Ignoruoti NėraDuomenų reikšmes	IGNORE_NODATA	[boolean] Numatytas: Taip	Neįjungus šios parinkties, visos įvesties sluoksnių NėraDuomenų celės reikš NėraDuomenų celes išvesties rastre
Atskaitos sluoksnis	REFERENCE_LAYE	[rastras]	Išvesties sluoksnio kūrimo atskaitos sluoksnis (apimtis, CRS, taškų dimensijos)
Išvesties sluoksnis	OUTPUT	[toks pat, kaip įvestis] Numatytas: [Įrašyti į laikiną failą]	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties NėraDuomenų reikšmė	OUTPUT_NODATA_	[skaičius: slankaus kablelio] Numatyta: -9999.0	NėraDuomenų reikšmė, kurią reikia naudoti išvesties sluoksnyje
Kūrimo parinktys Pasirinktinis	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	OUTPUT	[rastras]	Išvesties rastras su rezultatu
CRS identifikatorius	CRS_AUTHID	[tekstas]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	EXTENT	[tekstas]	Išvesties rastro erdvinė apimtis
Plotis taškais	WIDTH_IN_PIXEL	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius
Aukštis taškais	HEIGHT_IN_PIXE	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	TOTAL_PIXEL_CO	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius

Pythono kodas

Algoritmo ID: `native:cellstackpercentile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Cell stack percentrank from raster layer

Calculates the cell-wise percentrank value of a stack of rasters based on an input value raster and writes them to an output raster.

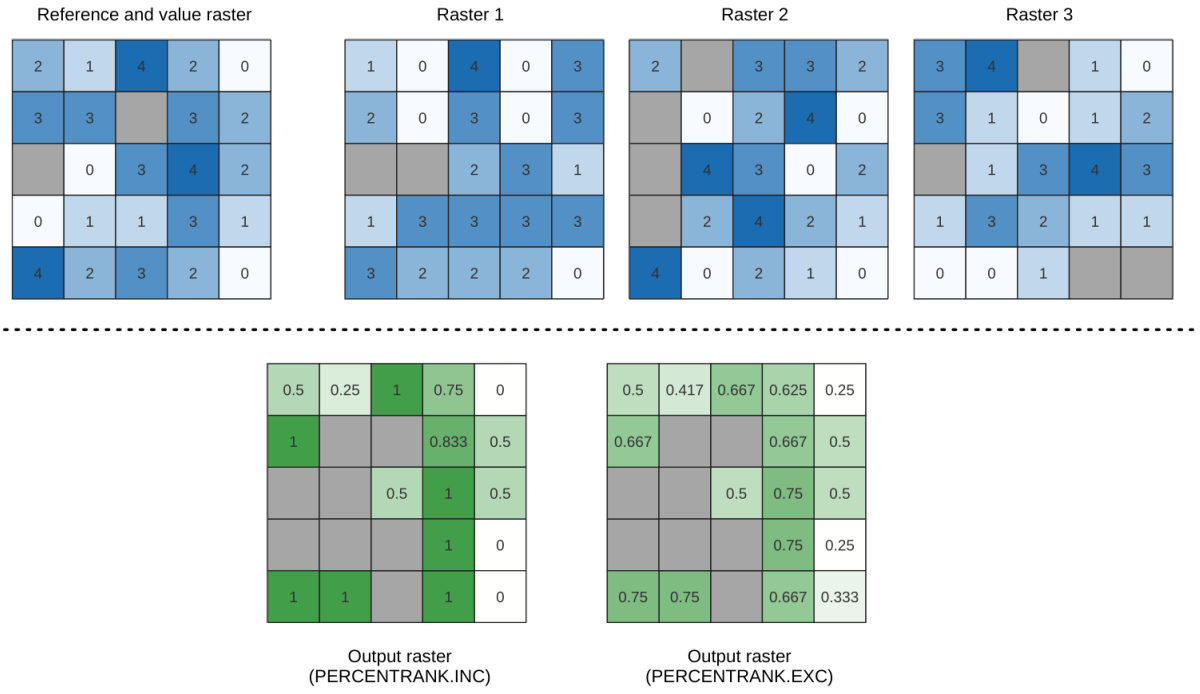
At each cell location, the current value of the value raster is ranked among the respective values in the stack of all overlaid and sorted cell values of the input rasters. For values outside of the the stack value distribution, the algorithm returns NoData because the value cannot be ranked among the cell values.

Yra du procentilių skaičiavimo metodai:

- Tiesinė interpoliacija imtinai (PERCENTRANK.INC)
- Tiesinė interpoliacija išimtinai (PERCENTRANK.EXC)

Tiesinės interpoliacijos metodai skirtingiems procentiliams grąžina unikalią reikšmę. Abu interpoliacijos metodai veikia pagal atitinkamus LibreOffice ar Microsoft Excel įgyvendinimus.

Išvesties rastro apimtis ir rezoliucija nustatoma pagal atskaitos rastrą. Įvesties rastro sluoksniai, kurie neatitinka atskaitos rastro sluoksnio celės dydžio, bus perrinkti naudojant artimiausio kaimyno metodą. Įvesties sluoksnių reikšmės NėraDuomenų reikš NėraDuomenų išvestį, jei nenurodytas parametras „Ignoruoti NėraDuomenų reikšmes“. Išvesties rastro duomenų tipas visada bus `Float32`.



24.51 Fig. : Ranking the value raster layer cells. `NoData` cells (grey) are ignored.

Taip pat žiūrėkite

Celių krūvos procentilis, *Celių krūvos procentinis rangas pagal reikšmę*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`INPUT`	[rastras] [sąrašas]	Vertinami rastro sluoksniai. Jei rastro duomenų krūvoje naudojami kelių juostų rastrai, algoritmas analizę visada vykdys pirmoje rastro juostoje
Value raster layer	`INPUT_VALUE_RA`	[rastras]	The layer to rank the values among the stack of all overlaid layers
Value raster band	`VALUE_RASTER_B`	[skaičius: sveikas skaičius] Numatytas: 1	Band of the „value raster layer“ to compare to
Metodas	`METHOD`	[sąrašas] Numatytas: 0	Percentilės skaičiavimo metodas: • 0 — Tiesinė interpoliacija imtinai (PERCENTRANK.INC) • 1 — Tiesinė interpoliacija išimtinai (PERCENTRANK.EXC)
Ignoruoti NėraDuomenų reikšmes	`IGNORE_NODATA`	[boolean] Numatytas: Taip	Neįjungus šios parinkties, visos įvesties sluoksnių NėraDuomenų celės reikš NėraDuomenų celes išvesties rastre
Atskaitos sluoksnis	`REFERENCE_LAYE`	[rastras]	Išvesties sluoksnio kūrimo atskaitos sluoksnis (apimtis, CRS, taškų dimensijos)
Išvesties sluoksnis	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties NėraDuomenų reikšmė	`OUTPUT_NODATA_`	[skaičius: slankaus kablelio] Numatyta: -9999.0	NėraDuomenų reikšmė, kurią reikia naudoti išvesties sluoksnyje

continues on next page

Table 24.308 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	`OUTPUT`	[rastras]	Išvesties rastras su rezultatu
CRS identifikatorius	`CRS_AUTHID`	[tekstas]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	`EXTENT`	[tekstas]	Išvesties rastro erdvinė apimtis
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius
Aukštis taškais	`HEIGHT_IN_PIXE`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	`TOTAL_PIXEL_CO`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius

Pythono kodas

Algorithm ID: `native:cellstackpercentrankfromrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Cell statistics

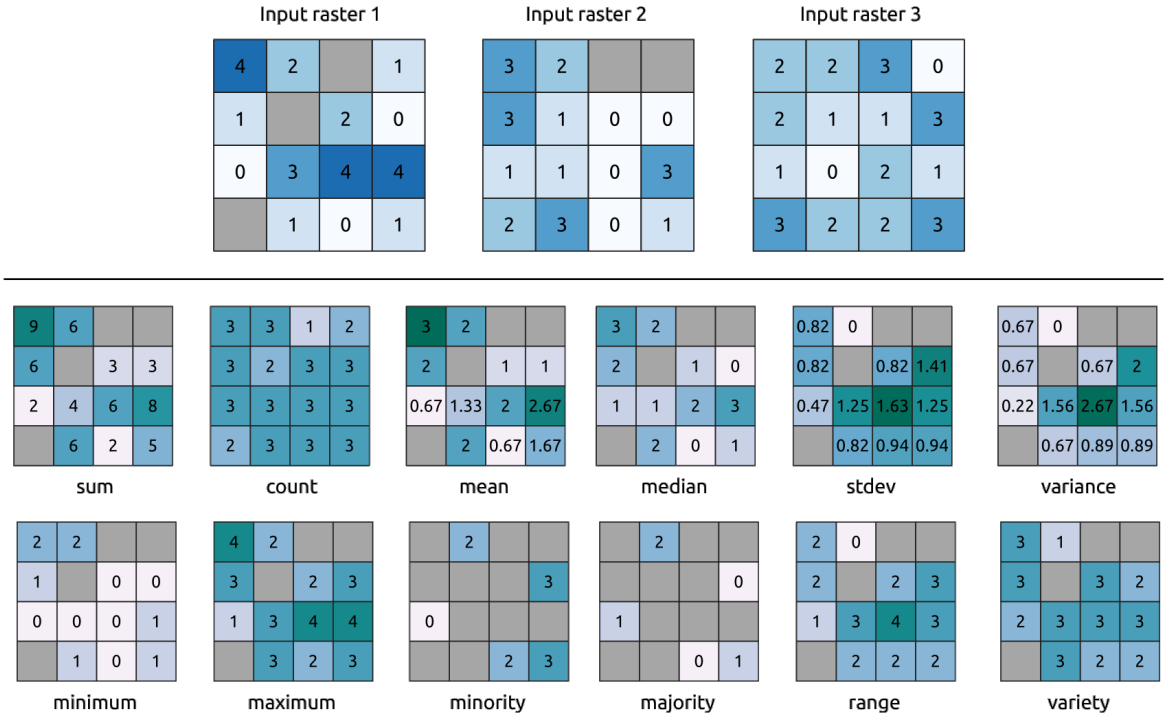
Computes per-cell statistics based on input raster layers and for each cell writes the resulting statistics to an output raster. At each cell location, the output value is defined as a function of all overlaid cell values of the input rasters.

By default, a NoData cell in ANY of the input layers will result in a NoData cell in the output raster. If the *Ignore NoData values* option is checked, then NoData inputs will be ignored in the statistic calculation. This may result in NoData output for locations where all cells are NoData.

The *Reference layer* parameter specifies an existing raster layer to use as a reference when creating the output raster. The output raster will have the same extent, CRS, and pixel dimensions as this layer.

Calculation details: Input raster layers that do not match the cell size of the reference raster layer will be resampled using `nearest neighbor resampling`. The output raster data type will be set to the most complex data type present in the input datasets except when using the functions `Mean`, `Standard deviation` and `Variance` (data type is always `Float32` or `Float64` depending on input float type) or `Count` and `Variety` (data type is always `Int32`).

- `Count`: The count statistic will always result in the number of cells without NoData values at the current cell location.
- `Median`: If the number of input layers is even, the median will be calculated as the arithmetic mean of the two middle values of the ordered cell input values.
- `Minority/Majority`: If no unique minority or majority could be found, the result is NoData, except all input cell values are equal.



24.52 Fig. : Example with all the statistic functions. `NoData` cells (grey) are taken into account.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`INPUT`	[rastras] [sąrašas]	Įvesties rastro sluoksniai
Statistic	`STATISTIC`	[sąrašas] Numatytas: 0	Available statistics. Options: • 0 — Sum • 1 — Count • 2 — Mean • 3 — Median • 4 — Standard deviation • 5 — Variance • 6 — Minimum • 7 — Maksimalus • 8 — Minority (least common value) • 9 — Majority (most common value) • 10 — Range (max - min) • 11 — Variety (unique value count)

continues on next page

Table 24.310 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Ignoruoti NėraDuomenų reikšmes	`IGNORE_NODATA`	[boolean] Numatytas: Taip	Calculate statistics also for all cells stacks, ignoring NoData occurrence.
Atskaitos sluoksnis	`REF_LAYER`	[rastras]	The reference layer to create the output layer from (extent, CRS, pixel dimensions)
Išvesties sluoksnis	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties NėraDuomenų reikšmė Pasirinktinis	`OUTPUT_NO_DATA`	[skaičius: slankaus kablelio] Numatyta: -9999.0	NėraDuomenų reikšmė, kurią reikia naudoti išvesties sluoksnyje
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
CRS identifikatorius	`CRS_AUTHID`	[crs]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	`EXTENT`	[tekstas]	Išvesties rastro erdvinė apimtis
Aukštis taškais	`HEIGHT_IN_PIXE`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Išvesties rastras	`OUTPUT`	[rastras]	Išvesties rastras su rezultatu
Galutinis taškų skaičius	`TOTAL_PIXEL_CO`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius

Pythono kodas

Algorithm ID: `native:cellstatistics`

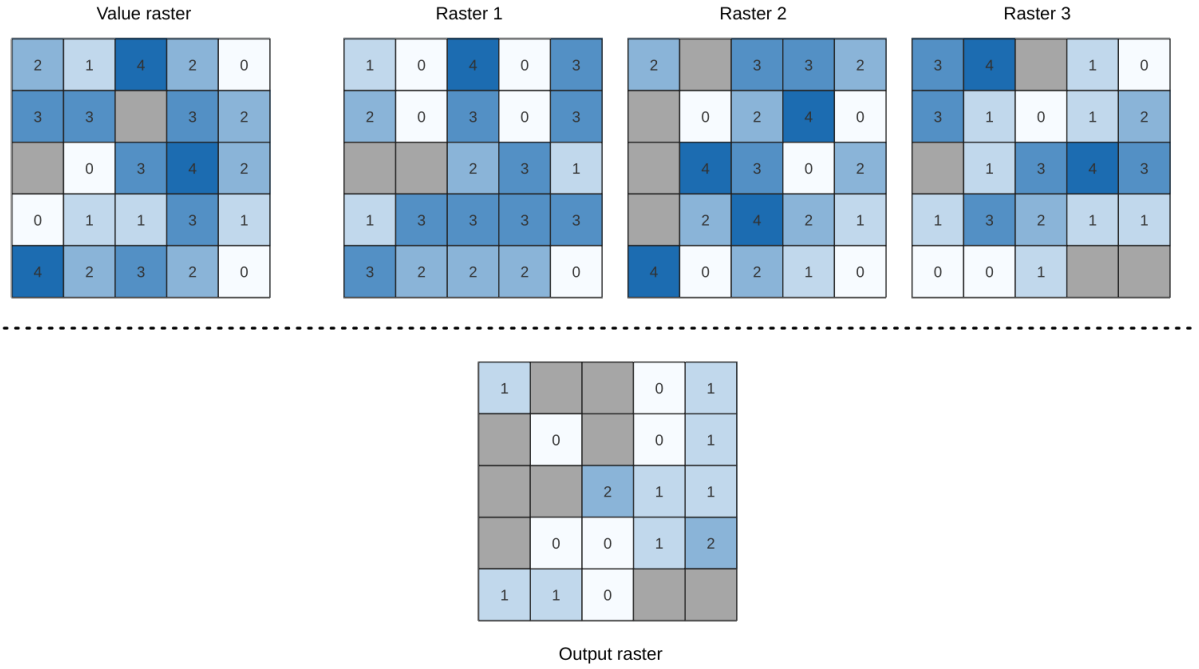
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Equal to frequency

Evaluates on a cell-by-cell basis the frequency (number of times) the values of an input stack of rasters are equal to the value of a value layer. The output raster extent and resolution are defined by the input raster layer and is always of `Int32` type.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. The output NoData value can be set manually.



24.53 Fig. : For each cell in the output raster, the value represents the number of times that the corresponding cells in the list of rasters are the same as the value raster. `NoData` cells (grey) are taken into account.

Taip pat žiūrėkite

Greater than frequency, *Less than frequency*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input value raster	`INPUT_VALUE_RA`	[rastras]	The input value layer serves as reference layer for the sample layers
Value raster band	`INPUT_VALUE_RA`	[rastro juosta] Default: The first band of the raster layer	Select the band you want to use as sample
Input raster layers	`INPUT_RASTERS`	[rastras] [sąrašas]	Vertinami rastro sluoksniai. Jei rastro duomenų krūvoje naudojami kelių juostų rastrai, algoritmas analizę visada vykdys pirmoje rastro juostoje
Ignoruoti NėraDuomenų reikšmes	`IGNORE_NODATA`	[boolean] Numatytas: ne	If unchecked, any NoData cells in the value raster or the data layer stack will result in a NoData cell in the output raster
Išvesties sluoksnis	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties NėraDuomenų reikšmė Pasirinktinis	`OUTPUT_NO_DATA`	[skaičius: slankaus kablelio] Numatyta: -9999.0	NėraDuomenų reikšmė, kurią reikia naudoti išvesties sluoksnyje
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	`OUTPUT`	[rastras]	Išvesties rastras su rezultatu
CRS identifikatorius	`CRS_AUTHID`	[tekstas]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	`EXTENT`	[tekstas]	Išvesties rastro erdvinė apimtis
Count of cells with equal value occurrences	`FOUND_LOCATION`	[skaičius: sveikas skaičius]	
Aukštis taškais	`HEIGHT_IN_PIXE`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	`TOTAL_PIXEL_CO`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius
Mean frequency at valid cell locations	`MEAN_FREQUENCY`	[skaičius: slankaus kablelio]	
Count of value occurrences	`OCCURRENCE_COU`	[skaičius: sveikas skaičius]	
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius

Pythono kodas

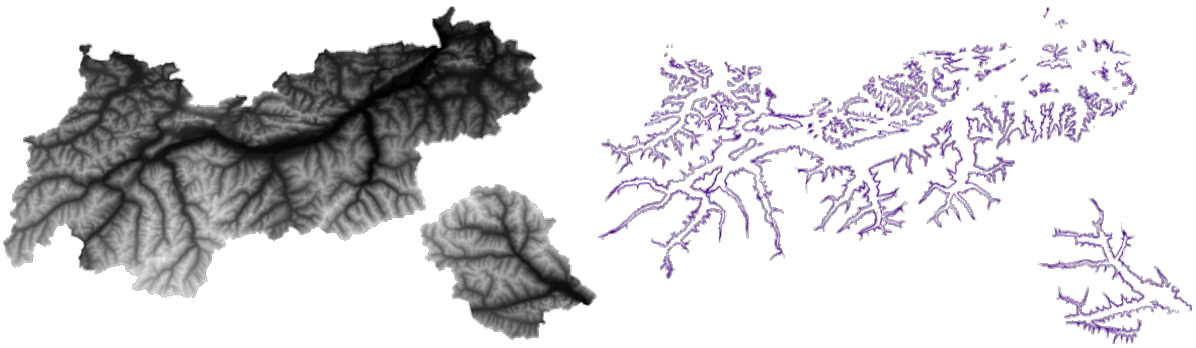
Algorithm ID: `native:equaltofrequency`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Fuzzify raster (gaussian membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Gaussian membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The gaussian membership function is defined as $\mu(x) = e^{-f1*(x-f2)^2}$, where *f1* is the spread and *f2* the midpoint.



24.54 Fig. : Fuzzify raster example. Input raster source: Land Tirol - data.tirol.gv.at.

Taip pat žiūrėkite

Fuzzify raster (large membership), *Fuzzify raster (linear membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (power membership)*, *Fuzzify raster (small membership)*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input Raster	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Band Number	`BAND`	[rastro juosta] Default: The first band of the raster layer	If the raster is multiband, choose the band that you want to fuzzify.
Function midpoint	`FUZZYMIDPOINT`	[skaičius: slankaus kablelio] Numatytas: 10.0	Midpoint of the gaussian function
Function spread	`FUZZYSPREAD`	[skaičius: slankaus kablelio] Default: 0.01	Spread of the gaussian function
Fuzzified raster	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ,‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Fuzzified raster	`OUTPUT`	[toks pat, kaip įvestis]	Išvesties rastras su rezultatu
CRS identifikatorius	`CRS_AUTHID`	[crs]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	`EXTENT`	[tekstas]	Išvesties rastro erdvinė apimtis
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius
Aukštis taškais	`HEIGHT_IN_PIXE`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	`TOTAL_PIXEL_CO`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius

Pythono kodas

Algorithm ID: `native:fuzzifyrastergaussianmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Fuzzify raster (large membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Large membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The large membership function is defined as $\mu(x) = \frac{1}{1 + \left(\frac{x}{f2}\right)^{-f1}}$, where *f1* is the spread and *f2* the midpoint.

Taip pat žiūrėkite

Fuzzify raster (gaussian membership), *Fuzzify raster (linear membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (power membership)*, *Fuzzify raster (small membership)*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input Raster	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Band Number	`BAND`	[rastro juosta] Default: The first band of the raster layer	If the raster is multiband, choose the band that you want to fuzzify.

continues on next page

Table 24.319 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Function midpoint	`FUZZYMIDPOINT`	[skaičius: slankaus kablelio] Numatytas: 50.0	Midpoint of the large function
Function spread	`FUZZYSPREAD`	[skaičius: slankaus kablelio] Default: 5.0	Spread of the large function
Fuzzified raster	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Fuzzified raster	`OUTPUT`	[toks pat, kaip įvestis]	Išvesties rastras su rezultatu
CRS identifikatorius	`CRS_AUTHID`	[crs]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	`EXTENT`	[tekstas]	Išvesties rastro erdvinė apimtis
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius
Aukštis taškais	`HEIGHT_IN_PIXE`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	`TOTAL_PIXEL_CO`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius

Pythono kodas

Algorithm ID: `native:fuzzifyrasterlargemembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Fuzzify raster (linear membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Linear membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The linear function is defined as $\mu(X)\begin{cases}0 & x \leq a \\ \frac{x-a}{b-a} & a < x < b \\ 1 & x \geq b\end{cases}$, where a is the low bound and b the high bound. This equation assigns membership values using a linear transformation for pixel values between the low and high bounds. Pixels values smaller than the low bound are given 0 membership whereas pixel values greater than the high bound are given 1 membership.

Taip pat žiūrėkite

Fuzzify raster (gaussian membership), *Fuzzify raster (large membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (power membership)*, *Fuzzify raster (small membership)*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input Raster	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Band Number	`BAND`	[rastro juosta] Default: The first band of the raster layer	If the raster is multiband, choose the band that you want to fuzzify.
Low fuzzy membership bound	`FUZZYLOWBOUND`	[skaičius: slankaus kablelio] Numatytas: 0.0	Low bound of the linear function
High fuzzy membership bound	`FUZZYHIGHBOUND`	[skaičius: slankaus kablelio] Numatytas: 1.0	High bound of the linear function
Fuzzified raster	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Fuzzified raster	`OUTPUT`	[toks pat, kaip įvestis]	Išvesties rastras su rezultatu
CRS identifikatorius	`CRS_AUTHID`	[crs]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	`EXTENT`	[tekstas]	Išvesties rastro erdvinė apimtis
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius
Aukštis taškais	`HEIGHT_IN_PIXE`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	`TOTAL_PIXEL_CO`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius

Pythono kodas

Algorithm ID: `native:fuzzifyrasterlinearmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Fuzzify raster (near membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Near membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The near membership function is defined as $\mu(x) = \frac{1}{1 + f1 * (x - f2)^2}$, where *f1* is the spread and *f2* the midpoint.

Taip pat žiūrėkite

Fuzzify raster (gaussian membership), *Fuzzify raster (large membership)*, *Fuzzify raster (linear membership)*, *Fuzzify raster (power membership)*, *Fuzzify raster (small membership)*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input Raster	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Band Number	`BAND`	[rastro juosta] Default: The first band of the raster layer	If the raster is multiband, choose the band that you want to fuzzify.
Function midpoint	`FUZZYMIDPOINT`	[skaičius: slankaus kablelio] Numatytas: 50.0	Midpoint of the near function
Function spread	`FUZZYSPREAD`	[skaičius: slankaus kablelio] Default: 0.01	Spread of the near function
Fuzzified raster	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Fuzzified raster	`OUTPUT`	[toks pat, kaip įvestis]	Išvesties rastras su rezultatu
CRS identifikatorius	`CRS_AUTHID`	[crs]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	`EXTENT`	[tekstas]	Išvesties rastro erdvinė apimtis

continues on next page

Table 24.327 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius
Aukštis taškais	`HEIGHT_IN_PIXE`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	`TOTAL_PIXEL_CO`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius

Pythono kodas

Algorithm ID: `native:fuzzifyrasternearmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Fuzzify raster (power membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Power membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The power function is defined as $\mu(x)\begin{cases}0 & x \le a \\ \left(\frac{x-a}{b-a}\right)^{f1} & a < x < b \\ 1 & x \ge b\end{cases}$, where a is the low bound, b is the high bound, and $f1$ the exponent. This equation assigns membership values using the power transformation for pixel values between the low and high bounds. Pixels values smaller than the low bound are given 0 membership whereas pixel values greater than the high bound are given 1 membership.

Taip pat žiūrėkite

Fuzzify raster (gaussian membership), *Fuzzify raster (large membership)*, *Fuzzify raster (linear membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (small membership)*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input Raster	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Band Number	`BAND`	[rastro juosta] Default: The first band of the raster layer	If the raster is multiband, choose the band that you want to fuzzify.
Low fuzzy membership bound	`FUZZYLOWBOUND`	[skaičius: slankaus kablelio] Numatytas: 0.0	Low bound of the power function

continues on next page

Table 24.328 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
High fuzzy membership bound	`FUZZYHIGHBOUND`	[skaičius: slankaus kablelio] Numatytas: 1.0	High bound of the power function
High fuzzy membership bound	`FUZZYEXPONENT`	[skaičius: slankaus kablelio] Numatyta: 2.0	Exponent of the power function
Fuzzified raster	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Fuzzified raster	`OUTPUT`	[toks pat, kaip įvestis]	Išvesties rastras su rezultatu
CRS identifikatorius	`CRS_AUTHID`	[crs]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	`EXTENT`	[tekstas]	Išvesties rastro erdvinė apimtis
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius
Aukštis taškais	`HEIGHT_IN_PIXE`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	`TOTAL_PIXEL_CO`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius

Pythono kodas

Algorithm ID: `native:fuzzifyrasterpowermembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Fuzzify raster (small membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Small membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The small membership function is defined as $\mu(x) = \frac{1}{1 + \left(\frac{x}{f2}\right)^{f1}}$, where *f1* is the spread and *f2* the midpoint.

Taip pat žiūrėkite

Fuzzify raster (gaussian membership), *Fuzzify raster (large membership)* *Fuzzify raster (linear membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (power membership)*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input Raster	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Band Number	`BAND`	[rastro juosta] Default: The first band of the raster layer	If the raster is multiband, choose the band that you want to fuzzify.
Function midpoint	`FUZZYMIDPOINT`	[skaičius: slankaus kablelio] Numatytas: 50.0	Midpoint of the small function
Function spread	`FUZZYSPREAD`	[skaičius: slankaus kablelio] Default: 5.0	Spread of the small function
Fuzzified raster	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Fuzzified raster	`OUTPUT`	[toks pat, kaip įvestis]	Išvesties rastras su rezultatu
CRS identifikatorius	`CRS_AUTHID`	[crs]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	`EXTENT`	[tekstas]	Išvesties rastro erdvinė apimtis
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius
Aukštis taškais	`HEIGHT_IN_PIXE`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	`TOTAL_PIXEL_CO`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius

Pythono kodas

Algorithm ID: `native:fuzzifyrastersmallmembership`

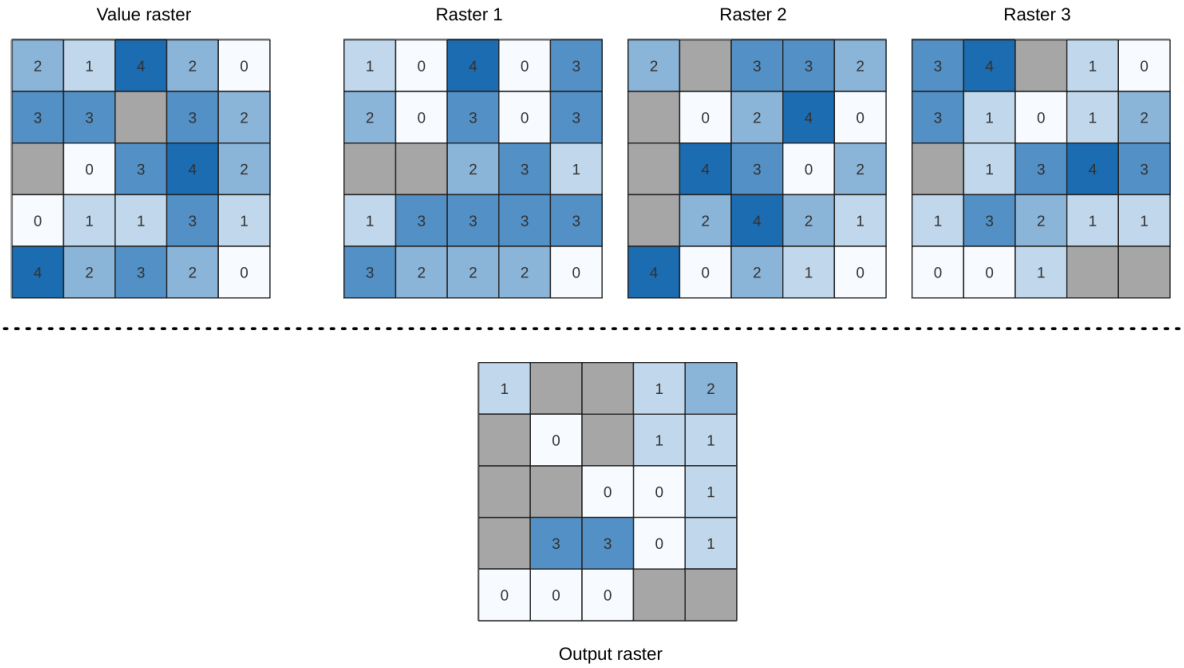
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Greater than frequency

Evaluates on a cell-by-cell basis the frequency (number of times) the values of an input stack of rasters are equal to the value of a value raster. The output raster extent and resolution is defined by the input raster layer and is always of `Int32` type.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. The output NoData value can be set manually.



24.55 Fig. : For each cell in the output raster, the value represents the number of times that the corresponding cells in the list of rasters are greater than the value raster. `NoData` cells (grey) are taken into account.

Taip pat žiūrėkite

Equal to frequency, *Less than frequency*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input value raster	`INPUT_VALUE_RA`	[rastras]	The input value layer serves as reference layer for the sample layers
Value raster band	`INPUT_VALUE_RA`	[rastro juosta] Default: The first band of the raster layer	Select the band you want to use as sample
Input raster layers	`INPUT_RASTERS`	[rastras] [sąrašas]	Vertinami rastro sluoksniai. Jei rastro duomenų krūvoje naudojami kelių juostų rastrai, algoritmas analizę visada vykdys pirmoje rastro juostoje
Ignoruoti NėraDuomenų reikšmes	`IGNORE_NODATA`	[boolean] Numatytas: ne	If unchecked, any NoData cells in the value raster or the data layer stack will result in a NoData cell in the output raster

continues on next page

Table 24.334 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	OUTPUT	[toks pat, kaip įvestis] Numatytas: [Įrašyti į laikiną failą]	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties NėraDuomenų reikšmė Pasirinktinis	OUTPUT_NO_DATA	[skaičius: slankaus kablelio] Numatyta: -9999.0	NėraDuomenų reikšmė, kurią reikia naudoti išvesties sluoksnyje
Kūrimo parinktys Pasirinktinis	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	OUTPUT	[rastras]	Išvesties rastras su rezultatu
CRS identifikatorius	CRS_AUTHID	[tekstas]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	EXTENT	[tekstas]	Išvesties rastro erdvinė apimtis
Count of cells with equal value occurrences	FOUND_LOCATION	[skaičius: sveikas skaičius]	
Aukštis taškais	HEIGHT_IN_PIXE	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	TOTAL_PIXEL_CO	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius
Mean frequency at valid cell locations	MEAN_FREQUENCY	[skaičius: slankaus kablelio]	
Count of value occurrences	OCCURRENCE_COU	[skaičius: sveikas skaičius]	
Plotis taškais	WIDTH_IN_PIXEL	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius

Pythono kodas

Algorithm ID: `native:greaterthanfrequency`

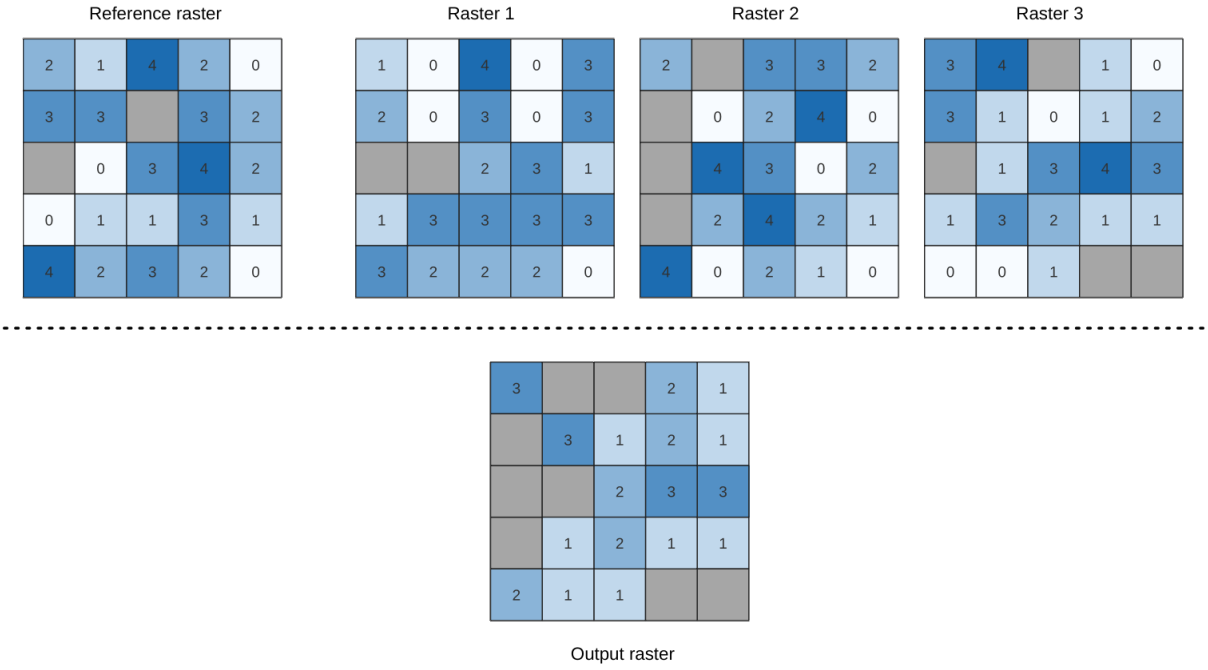
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Highest position in raster stack

Evaluates on a cell-by-cell basis the position of the raster with the highest value in a stack of rasters. Position counts start with 1 and range to the total number of input rasters. The order of the input rasters is relevant for the algorithm. If multiple rasters feature the highest value, the first raster will be used for the position value.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. Any NoData cells in the raster layer stack will result in a NoData cell in the output raster unless the „ignore NoData“ parameter is checked. The output NoData value can be set manually. The output rasters extent and resolution is defined by a reference raster layer and is always of `Int32` type.



Taip pat žiūrėkite

Lowest position in raster stack

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input raster layers	`INPUT_RASTERS`	[rastras] [sąrašas]	List of raster layers to compare with
Atskaitos sluoksnis	`REFERENCE_LAYE`	[rastras]	Išvesties sluoksnio kūrimo atskaitos sluoksnis (apimtis, CRS, taškų dimensijos)
Ignoruoti NėraDuomenų reikšmes	`IGNORE_NODATA`	[boolean] Numatytas: ne	If unchecked, any NoData cells in the data layer stack will result in a NoData cell in the output raster
Išvesties sluoksnis	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specification of the output raster containing the result. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties NėraDuomenų reikšmė	`OUTPUT_NODATA_`	[skaičius: slankaus kablelio] Numatyta: -9999.0	NėraDuomenų reikšmė, kurią reikia naudoti išvesties sluoksnyje
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	`OUTPUT`	[rastras]	Išvesties rastras su rezultatu
CRS identifikatorius	`CRS_AUTHID`	[tekstas]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	`EXTENT`	[tekstas]	Išvesties rastro erdvinė apimtis
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius
Aukštis taškais	`HEIGHT_IN_PIXE`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	`TOTAL_PIXEL_CO`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius

Pythono kodas

Algorithm ID: `native:highestpositioninrasterstack`

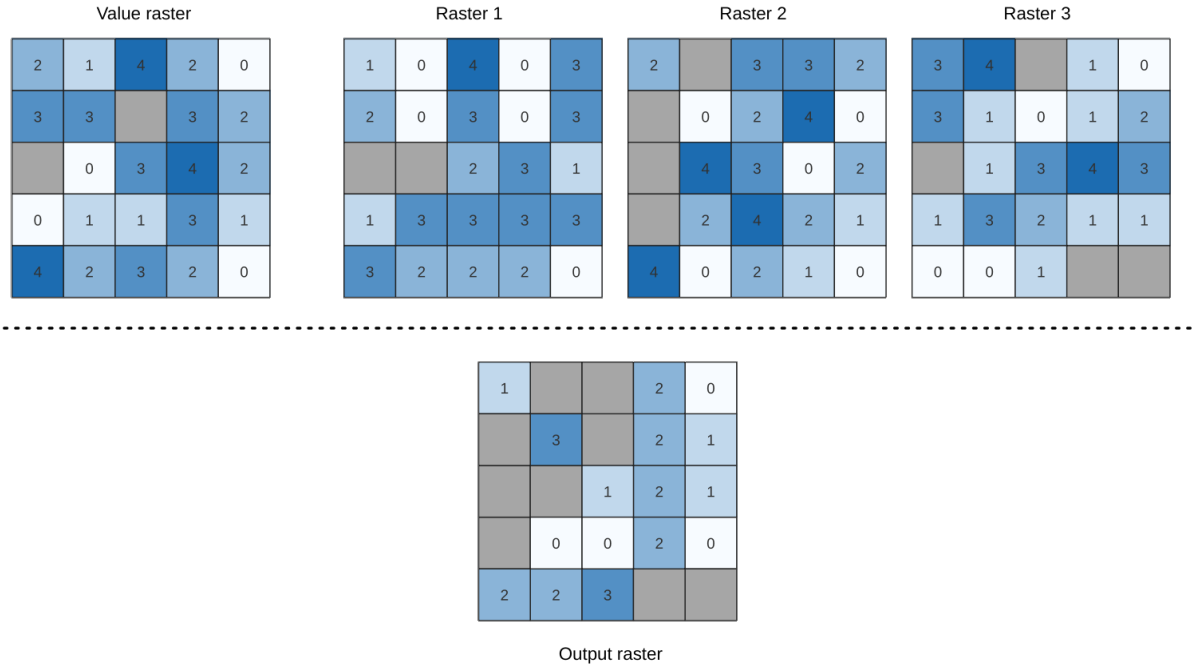
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Less than frequency

Evaluates on a cell-by-cell basis the frequency (number of times) the values of an input stack of rasters are less than the value of a value raster. The output raster extent and resolution is defined by the input raster layer and is always of `Int32` type.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. The output NoData value can be set manually.



24.56 Fig. : For each cell in the output raster, the value represents the number of times that the corresponding cells in the list of rasters are less than the value raster. `NoData` cells (grey) are taken into account.

Taip pat žiūrėkite

Equal to frequency, *Greater than frequency*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input value raster	`INPUT_VALUE_RA`	[rastras]	The input value layer serves as reference layer for the sample layers
Value raster band	`INPUT_VALUE_RA`	[rastro juosta] Default: The first band of the raster layer	Select the band you want to use as sample
Input raster layers	`INPUT_RASTERS`	[rastras] [sąrašas]	Vertinami rastro sluoksniai. Jei rastro duomenų krūvoje naudojami kelių juostų rastrai, algoritmas analizę visada vykdys pirmoje rastro juostoje
Ignoruoti NėraDuomenų reikšmes	`IGNORE_NODATA`	[boolean] Numatytas: ne	If unchecked, any NoData cells in the value raster or the data layer stack will result in a NoData cell in the output raster
Išvesties sluoksnis	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties NėraDuomenų reikšmė Pasirinktinis	`OUTPUT_NO_DATA`	[skaičius: slankaus kablelio] Numatyta: -9999.0	NėraDuomenų reikšmė, kurią reikia naudoti išvesties sluoksnyje
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	`OUTPUT`	[rastras]	Išvesties rastras su rezultatu
CRS identifikatorius	`CRS_AUTHID`	[tekstas]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	`EXTENT`	[tekstas]	Išvesties rastro erdvinė apimtis
Count of cells with equal value occurrences	`FOUND_LOCATION`	[skaičius: sveikas skaičius]	
Aukštis taškais	`HEIGHT_IN_PIXE`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	`TOTAL_PIXEL_CO`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius
Mean frequency at valid cell locations	`MEAN_FREQUENCY`	[skaičius: slankaus kablelio]	
Count of value occurrences	`OCCURRENCE_COU`	[skaičius: sveikas skaičius]	
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius

Pythono kodas

Algorithm ID: `native:lessthanfrequency`

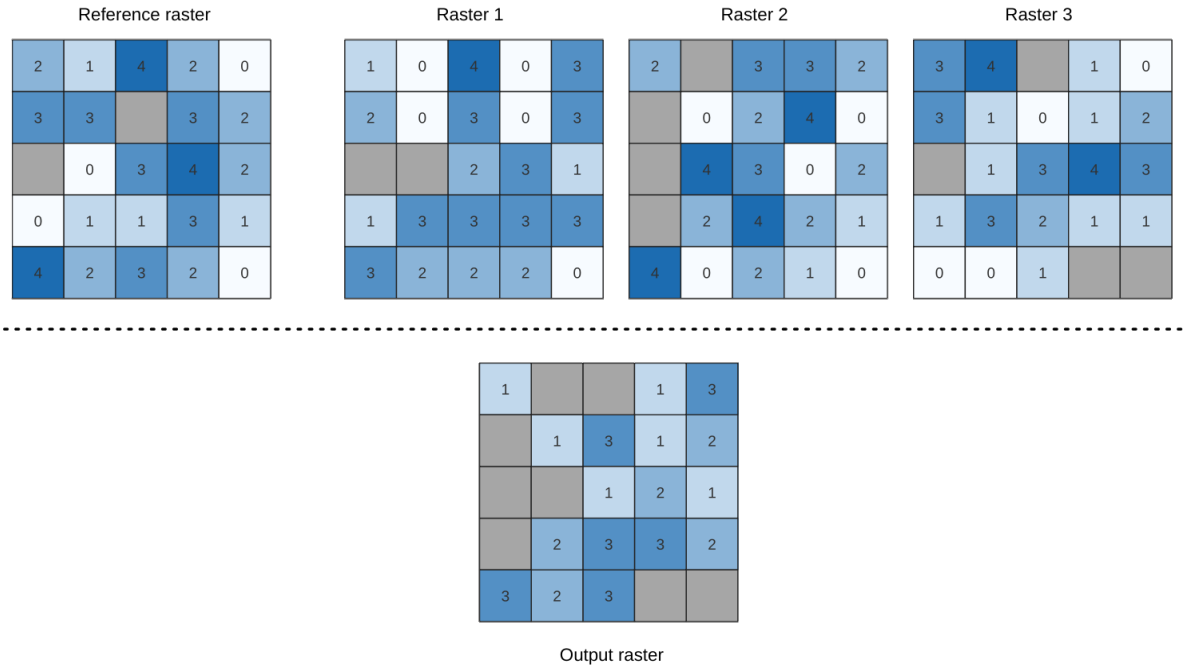
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Lowest position in raster stack

Evaluates on a cell-by-cell basis the position of the raster with the lowest value in a stack of rasters. Position counts start with 1 and range to the total number of input rasters. The order of the input rasters is relevant for the algorithm. If multiple rasters feature the lowest value, the first raster will be used for the position value.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. Any NoData cells in the raster layer stack will result in a NoData cell in the output raster unless the „ignore NoData“ parameter is checked. The output NoData value can be set manually. The output rasters extent and resolution is defined by a reference raster layer and is always of `Int32` type.



Taip pat žiūrėkite

Highest position in raster stack

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input raster layers	`INPUT_RASTERS`	[rastras] [sąrašas]	List of raster layers to compare with
Atskaitos sluoksnis	`REFERENCE_LAYE`	[rastras]	Išvesties sluoksnio kūrimo atskaitos sluoksnis (apimtis, CRS, taškų dimensijos)
Ignoruoti NėraDuomenų reikšmes	`IGNORE_NODATA`	[boolean] Numatytas: ne	If unchecked, any NoData cells in the data layer stack will result in a NoData cell in the output raster
Išvesties sluoksnis	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specification of the output raster containing the result. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties NėraDuomenų reikšmė	OUTPUT_NODATA_	[skaičius: slankaus kablelio] Numatyta: -9999.0	NėraDuomenų reikšmė, kurią reikia naudoti išvesties sluoksnyje
Kūrimo parinktys Pasirinktinis	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	OUTPUT	[rastras]	Išvesties rastras su rezultatu
CRS identifikatorius	CRS_AUTHID	[tekstas]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	EXTENT	[tekstas]	Išvesties rastro erdvinė apimtis
Plotis taškais	WIDTH_IN_PIXEL	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius
Aukštis taškais	HEIGHT_IN_PIXE	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	TOTAL_PIXEL_CO	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius

Pythono kodas

Algorithm ID: `native:lowestpositioninrasterstack`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster boolean AND

Calculates the boolean `AND` for a set of input rasters. If all of the input rasters have a non-zero value for a pixel, that pixel will be set to `1` in the output raster. If any of the input rasters have `0` values for the pixel it will be set to `0` in the output raster.

The reference layer parameter specifies an existing raster layer to use as a reference when creating the output raster. The output raster will have the same extent, CRS, and pixel dimensions as this layer.

By default, a NoData pixel in ANY of the input layers will result in a NoData pixel in the output raster. If the *Treat NoData values as false* option is checked, then NoData inputs will be treated the same as a `0` input value.

Taip pat žiūrėkite

Raster boolean OR

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`INPUT`	[rastras] [sąrašas]	List of input raster layers
Atskaitos sluoksnis	`REF_LAYER`	[rastras]	The reference layer to create the output layer from (extent, CRS, pixel dimensions)
Treat NoData values as false	`NODATA_AS_FALS`	[boolean] Numatytas: ne	Treat NoData values in the input files as 0 when performing the operation
Išvesties sluoksnis	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specification of the output raster containing the result. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties NėraDuomenų reikšmė	`NO_DATA`	[skaičius: slankaus kablelio] Numatyta: -9999.0	NėraDuomenų reikšmė, kurią reikia naudoti išvesties sluoksnyje

continues on next page

Table 24.347 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties duomenų tipas	DATA_TYPE	[sąrašas] Numatytas:	Output raster data type. Options: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — Int32 (Thirty two bit signed integer (qint32)) • 4 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)
Kūrimo parinktys Pasirinktinis	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Apimtis	EXTENT	[tekstas]	Išvesties rastro erdvinė apimtis
CRS identifikatorius	CRS_AUTHID	[crs]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Plotis taškais	WIDTH_IN_PIXEL	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius
Aukštis taškais	HEIGHT_IN_PIXE	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	TOTAL_PIXEL_CO	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius
NoData pixel count	NODATA_PIXEL_C	[skaičius: sveikas skaičius]	The count of NoData pixels in the output raster layer

continues on next page

Table 24.348 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
True pixel count	TRUE_PIXEL_COU	[skaičius: sveikas skaičius]	The count of True pixels (value = 1) in the output raster layer
False pixel count	FALSE_PIXEL_CO	[skaičius: sveikas skaičius]	The count of False pixels (value = 0) in the output raster layer
Išvesties sluoksnis	OUTPUT	[rastras]	Išvesties rastras su rezultatu

Pythono kodas

Algorithm ID: `native:rasterbooleanand`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster boolean OR

Calculates the boolean OR for a set of input rasters. If all of the input rasters have a zero value for a pixel, that pixel will be set to 0 in the output raster. If any of the input rasters have 1 values for the pixel it will be set to 1 in the output raster.

The reference layer parameter specifies an existing raster layer to use as a reference when creating the output raster. The output raster will have the same extent, CRS, and pixel dimensions as this layer.

By default, a NoData pixel in ANY of the input layers will result in a NoData pixel in the output raster. If the *Treat NoData values as false* option is checked, then NoData inputs will be treated the same as a 0 input value.

Taip pat žiūrėkite

Raster boolean AND

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	INPUT	[rastras] [sąrašas]	List of input raster layers
Atskaitos sluoksnis	REF_LAYER	[rastras]	The reference layer to create the output layer from (extent, CRS, pixel dimensions)
Treat NoData values as false	NODATA_AS_FALS	[boolean] Numatytas: ne	Treat NoData values in the input files as 0 when performing the operation
Išvesties sluoksnis	OUTPUT	[rastras] Numatytas: [Įrašyti į laikiną failą]	Specification of the output raster containing the result. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties NėraDuomenų reikšmė	`NO_DATA`	[skaičius: slankaus kablelio] Numatyta: -9999.0	NėraDuomenų reikšmė, kurią reikia naudoti išvesties sluoksnyje
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas:	Output raster data type. Options: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — Int32 (Thirty two bit signed integer (qint32)) • 4 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Apimtis	`EXTENT`	[tekstas]	Išvesties rastro erdvinė apimtis
CRS identifikatorius	`CRS_AUTHID`	[crs]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius

continues on next page

Table 24.351 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Aukštis taškais	HEIGHT_IN_PIXE	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
Galutinis taškų skaičius	TOTAL_PIXEL_CO	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius
NoData pixel count	NODATA_PIXEL_C	[skaičius: sveikas skaičius]	The count of NoData pixels in the output raster layer
True pixel count	TRUE_PIXEL_COU	[skaičius: sveikas skaičius]	The count of True pixels (value = 1) in the output raster layer
False pixel count	FALSE_PIXEL_CO	[skaičius: sveikas skaičius]	The count of False pixels (value = 0) in the output raster layer
Išvesties sluoksnis	OUTPUT	[rastras]	Išvesties rastras su rezultatu

Pythono kodas

Algorithm ID: `native:rasterbooleanor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster calculator

Performs algebraic operations using raster layers.

The resulting layer will have its values computed according to an expression. The expression can contain numerical values, operators and references to any of the layers in the current project.

Taip pat žiūrėkite

Raster calculator (virtual), *Raster calculator*, *Raster Calculator*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	INPUT	[rastras] [sąrašas]	List of input raster layers
Išraiška	EXPRESSION	[išraiška]	*Raster-based expression* that will be used to calculate the output raster layer.

continues on next page

Table 24.352 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Output extent Pasirinktinis	`EXTENT`	[apimtis]	Specify the spatial extent of the output raster layer. If the extent is not specified, the minimum extent that covers all the selected reference layers will be used. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Output cell size (leave empty to set automatically) Pasirinktinis	`CELLSIZE`	[skaičius: slankaus kablelio]	Cell size of the output raster layer. If the cell size is not specified, the minimum cell size of the selected reference layer(s) will be used. The cell size will be the same for the X and Y axes.
Išvesties CRS Pasirinktinis	`CRS`	[crs]	CRS of the output raster layer. If the output CRS is not specified, the CRS of the first reference layer will be used.
Calculated	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (`|`).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Calculated	`OUTPUT`	[rastras]	Output raster file with the calculated values.

Pythono kodas

Algorithm ID: `native:rastercalc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster calculator (virtual)

Performs algebraic operations using raster layers and generates in-memory result.

The resulting layer will have its values computed according to an expression. The expression can contain numerical values, operators and references to any of the layers in the current project.

A virtual raster layer is a raster layer defined by its URI and whose pixels are calculated on-the-fly. It's not a new file on disk; the virtual layer is still connected to the rasters used in the calculation meaning that deleting or moving these rasters would break it. A *Layer name* can be provided, otherwise the calculation expression is used as such. Removing the virtual layer from the project deletes it, and it can be made persistent in file using the layer *Export* ► *Save as…* contextual menu.

Taip pat žiūrėkite

Raster calculator, *Raster calculator*, *Raster Calculator*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`INPUT`	[rastras] [sąrašas]	List of input raster layers
Išraiška	`EXPRESSION`	[išraiška]	*Raster-based expression* that will be used to calculate the output raster layer.

continues on next page

Table 24.355 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Output extent Pasirinktinis	`EXTENT`	[apimtis]	Specify the spatial extent of the output raster layer. If the extent is not specified, the minimum extent that covers all the selected reference layers will be used. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Output cell size (leave empty to set automatically) Pasirinktinis	`CELLSIZE`	[skaičius: slankaus kablelio]	Cell size of the output raster layer. If the cell size is not specified, the minimum cell size of the selected reference layer(s) will be used. The cell size will be the same for the X and Y axes.
Išvesties CRS Pasirinktinis	`CRS`	[crs]	CRS of the output raster layer. If the output CRS is not specified, the CRS of the first reference layer will be used.
Output layer name Pasirinktinis	`LAYER_NAME`	[tekstas]	The name to assign to the generated layer. If not set, the text of the calculation expression is used.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Calculated	`OUTPUT`	[rastras]	Output virtual raster layer with the calculated values.

Pythono kodas

Algorithm ID: `native:virtualrastercalc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster layer properties

Returns basic properties of the given raster layer, including the extent, size in pixels and dimensions of pixels (in map units), number of bands, and NoData value.

This algorithm is intended for use as a means of extracting these useful properties to use as the input values to other algorithms in a model - e.g. to allow to pass an existing raster's pixel sizes over to a GDAL raster algorithm.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Juostos numeris Pasirinktinis	`BAND`	[rastro juosta] Numatytas: jokio	Whether to also return properties of a specific band. If a band is specified, the noData value for the selected band is also returned.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Number of bands in raster	`BAND_COUNT`	[skaičius: sveikas skaičius]	The number of bands in the raster
CRS identifikatorius	`CRS_AUTHID`	[tekstas]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	`EXTENT`	[tekstas]	The raster layer extent in the CRS
Band has a NoData value set	`HAS_NODATA_VAL`	[Boolean]	Indicates whether the raster layer has a value set for NoData pixels in the selected band
Aukštis taškais	`HEIGHT_IN_PIXE`	[skaičius: sveikas skaičius]	The number of columns in the raster layer
Band NoData value	`NODATA_VALUE`	[skaičius: slankaus kablelio]	The value (if set) of the NoData pixels in the selected band
Pixel size (height) in map units	`PIXEL_HEIGHT`	[skaičius: sveikas skaičius]	Vertical size in map units of the pixel
Pixel size (width) in map units	`PIXEL_WIDTH`	[skaičius: sveikas skaičius]	Horizontal size in map units of the pixel
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	The number of rows in the raster layer
Maximum x-coordinate	`X_MAX`	[skaičius: slankaus kablelio]	
Minimum x-coordinate	`X_MIN`	[skaičius: slankaus kablelio]	
Maximum y-coordinate	`Y_MAX`	[skaičius: slankaus kablelio]	
Minimum y-coordinate	`Y_MIN`	[skaičius: slankaus kablelio]	

Pythono kodas

Algorithm ID: `native:rasterlayerproperties`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster layer statistics

Calculates basic statistics from the values in a given band of the raster layer. The output is loaded in the *Processing* ► *Results viewer* menu.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta] Numatytas: pirma įvesties sluoksnio juosta	If the raster is multiband, choose the band you want to get statistics for.
Statistika	`OUTPUT_HTML_FI`	[html] Numatytas: `[Įrašyti į laikiną failą]`	Specification of the output file: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Maksimali reikšmė	`MAX`	[skaičius: slankaus kablelio]	
Vidutinė reikšmė	`MEAN`	[skaičius: slankaus kablelio]	
Minimali reikšmė	`MIN`	[skaičius: slankaus kablelio]	

continues on next page

Table 24.360 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Statistika	`OUTPUT_HTML_FI`	[html]	The output file contains the following information: • Analyzed file: path of the raster layer • Minimum value: minimum value of the raster • Maximum value: maximum value of the raster • Range: difference between the maximum and minimum values • Sum: total sum of the values • Mean value: mean of the values • Standard deviation: standard deviation of the values • Sum of the squares: sum of the squared differences of each observation from the overall mean
Diapazonas	`RANGE`	[skaičius: slankaus kablelio]	
Standartinis nuokrypis	`STD_DEV`	[skaičius: slankaus kablelio]	
Suma	`SUM`	[skaičius: slankaus kablelio]	
Sum of the squares	`SUM_OF_SQUARES`	[skaičius: slankaus kablelio]	

Pythono kodas

Algorithm ID: `native:rasterlayerstatistics`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster layer unique values report

Returns the count and area of each unique value in a given raster layer. The calculation of the area is done in the area unit of the layer's CRS.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta] Numatytas: pirma įvesties sluoksnio juosta	If the raster is multiband, choose the band you want to get statistics for.

continues on next page

Table 24.361 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Unique values report	OUTPUT_HTML_FI	[file] Numatytas: [Įrašyti į laikiną failą]	Specification of the output file: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…
Unique values table	OUTPUT_TABLE	[vektorius: lentelė] Numatytas: [Praleisti išvestį]	Specification of the table for unique values: • Praleisti išvestį • Kurti laikiną sluoksnį (TEMPORARY_OUTPUT) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
CRS identifikatorius	CRS_AUTHID	[tekstas]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	EXTENT	[tekstas]	Išvesties rastro erdvinė apimtis
Aukštis taškais	HEIGHT_IN_PIXE	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
NoData pixel count	NODATA_PIXEL_C	[skaičius: sveikas skaičius]	The number of NoData pixels in the output raster layer
Galutinis taškų skaičius	TOTAL_PIXEL_CO	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius
Unique values report	OUTPUT_HTML_FI	[html]	The output HTML file contains the following information: • Analyzed file: the path of the raster layer • Extent: xmin, ymin, xmax, ymax coordinates of the extent • Projection: projection of the layer • Width in pixels: number of columns and pixel width size • Height in pixels: number of rows and pixel width size • Total pixel count: count of all the pixels • NoData pixel count: count of pixels with NoData value
Unique values table	OUTPUT_TABLE	[vektorius: lentelė]	A table with three columns: • *value*: pixel value • *count*: count of pixels with this value • *m2* or *deg2* or *ft2* or … : total area of pixels with this value. The column name depends on the area unit of the layer's CRS and the calculation is done in that unit.

continues on next page

Table 24.362 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Plotis taškais	`WIDTH_IN_PIXEL`	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius

Pythono kodas

Algorithm ID: `native:rasterlayeruniquevaluesreport`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster layer zonal statistics

Calculates statistics for a raster layer's values, categorized by zones defined in another raster layer.

Taip pat žiūrėkite

Zonal statistics

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta] Default: The first band of the raster layer	If the raster is multiband choose the band for which you want to calculate the statistics.
Zones layer	`ZONES`	[rastras]	Raster layer defining zones. Zones are given by contiguous pixels having the same pixel value.
Zones band number	`ZONES_BAND`	[rastro juosta] Default: The first band of the raster layer	If the raster is multiband, choose the band that defines the zones
Statistika	`OUTPUT_TABLE`	[vektorius: lentelė] Numatytas: `[Kurti laikiną sluoksnį]`	Specification of the output report. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Atskaitos sluoksnis Pasirinktinis	REF_LAYER	[sąrašas] Numatytas: 0	Raster layer used to calculate the centroids that will be used as reference when determining the zones in the output layer. One of: • 0 — Input layer: zones are determined by sampling the zone raster layer value at the centroid of each pixel from the source raster layer • 1 — Zones layer: the input raster layer will be sampled at the centroid of each pixel from the zones raster layer

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
CRS identifikatorius	CRS_AUTHID	[tekstas]	Išvesties rastro sluoksnio koordinačių atskaitos sistema
Apimtis	EXTENT	[tekstas]	Išvesties rastro erdvinė apimtis
Aukštis taškais	HEIGHT_IN_PIXE	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio eilučių skaičius
NoData pixel count	NODATA_PIXEL_C	[skaičius: sveikas skaičius]	The number of NoData pixels in the output raster layer
Statistika	OUTPUT_TABLE	[vektorius: lentelė]	The output layer contains the following information **for each zone**: • Area: the area in square raster units in the zone; • Sum: the total sum of the pixel values in the zone; • Count: the number of pixels in the zone; • Min: the minimum pixel value in the zone; • Max: the maximum pixel value in the zone; • Mean: the mean of the pixel values in the zone;
Galutinis taškų skaičius	TOTAL_PIXEL_CO	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio taškų skaičius
Plotis taškais	WIDTH_IN_PIXEL	[skaičius: sveikas skaičius]	Išvesties rastro sluoksnio stulpelių skaičius

Pythono kodas

Algorithm ID: `native:rasterlayerzonalstats`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster minimum/maximum

Extracts the minimum and maximum pixel values from a specified band of a raster layer. If multiple pixels share the same minimum or maximum value, only one of them will be included in the output.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Raster layer from which to extract the minimum and maximum pixel values.
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	The band of the raster to analyze. If the raster is multiband, specify the band number (starting from 1).
Extract	`EXTREMA`	[sąrašas] Default: 0 (Minimum and Maximum)	Choose which extrema to extract: • 0: Minimum and Maximum • 1: Minimum only • 2: Maximum only
Išvesties sluoksnis	`OUTPUT`	[vector: point] Numatytas: `[Kurti laikiną sluoksnį]`	Specification of the output layer. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	`OUTPUT`	[vector: point]	Vector layer with point features at the locations of the minimum and/or maximum pixel values.

Pythono kodas

Algorithm ID: `native:rasterminmax`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster rank

`NEW in 3.44`

Performs a cell-by-cell analysis in which output values match the rank of a sorted list of overlapping cell values from input layers. The output raster will be multi-band if multiple ranks are provided. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters.

Taip pat žiūrėkite

Celių krūvos procentinis rangas pagal reikšmę, *Celių krūvos procentilis*, *Cell stack percentrank from raster layer*, *Cell statistics*, *Raster calculator*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT_RASTERS`	[rastras] [sąrašas]	Input raster layer(s) to calculate the rank from.
Rank	`RANKS`	[numeric: integer] [list] Default: [1]	List of ranks to calculate for each pixel in the sorted list of cells values from the input rasters. Cells whose values match the given rank(s) are kept in the output layer. The ranks are 1-based, meaning that the first rank is 1. Negative values are also supported.
NoData value handling	`NODATA_HANDLIN`	[sąrašas] Numatytas: 0	Choose how to handle NoData values in the input raster: • 0 — Exclude NoData from values lists: will not include NoData values in the list of pixel values when calculating ranks. • 1 — Presence of NoData in a values list results in NoData output cells: the output raster will have NoData values in the corresponding locations.

continues on next page

Table 24.368 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Ranked	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specification of the output raster containing the rank values. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Output extent Pasirinktinis	`EXTENT`	[apimtis]	Specify the spatial extent of the output raster layer. If the extent is not specified, the minimum extent that covers all the selected reference layers will be used. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Output cell size Pasirinktinis	`CELL_SIZE`	[skaičius: slankaus kablelio]	Cell size of the output raster layer. If the cell size is not specified, the minimum cell size of the selected reference layer(s) will be used. The cell size will be the same for the X and Y axes.
Išvesties CRS Pasirinktinis	`CRS`	[crs]	CRS of the output raster layer. If the output CRS is not specified, the CRS of the first reference layer will be used.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Ranked	`OUTPUT`	[rastras]	Output raster layer containing the result.

Pythono kodas

Algorithm ID: `native:rasterrank`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster surface volume

Calculates the volume under a raster surface relative to a given base level. This is mainly useful for Digital Elevation Models (DEM).

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
INPUT layer	`INPUT`	[rastras]	Input raster, representing a surface
Juostos numeris	`BAND`	[rastro juosta] Default: The first band of the raster layer	If the raster is multiband, choose the band that shall define the surface.
Base level	`LEVEL`	[skaičius: slankaus kablelio] Numatytas: 0.0	Define a base or reference value. This base is used in the volume calculation according to the `Method` parameter (see below).
Metodas	`METHOD`	[sąrašas] Numatytas: 0	Define the method for the volume calculation given by the difference between the raster pixel value and the `Base level`. Options: • 0 — Count Only Above Base Level: only pixels above the base level will add to the volume. • 1 — Count Only Below Base Level: only pixels below the base level will add to the volume. • 2 — Subtract Volumes Below Base level: pixels above the base level will add to the volume, pixels below the base level will subtract from the volume. • 3 — Add Volumes Below Base level: Add the volume regardless whether the pixel is above or below the base level. This is equivalent to sum the absolute values of the difference between the pixel value and the base level.

continues on next page

Table 24.371 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Surface volume report	OUTPUT_HTML_FI	[html] Numatytas: [Įrašyti į laikiną failą]	Specification of the output HTML report. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (TEMPORARY_OUTPUT) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Surface volume table	OUTPUT_TABLE	[vektorius: lentelė] Numatytas: [Praleisti išvestį]	Specification of the output table. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (TEMPORARY_OUTPUT) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Volume	VOLUME	[skaičius: slankaus kablelio]	The calculated volume
Area	AREA	[skaičius: slankaus kablelio]	The area in square map units
Pixel_count	PIXEL_COUNT	[skaičius: sveikas skaičius]	The total number of pixels that have been analyzed
Surface volume report	OUTPUT_HTML_FI	[html]	The output report (containing volume, area and pixel count) in HTML format
Surface volume table	OUTPUT_TABLE	[vektorius: lentelė]	The output table (containing volume, area and pixel count)

Pythono kodas

Algorithm ID: `native:rastersurfacevolume`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Reclassify by layer

Reclassifies a raster band by assigning new class values based on the ranges specified in a vector table.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Rastro sluoksnis	`INPUT_RASTER`	[rastras]	Raster layer to reclassify
Juostos numeris	`RASTER_BAND`	[rastro juosta] Default: The first band of the raster layer	If the raster is multiband, choose the band you want to reclassify.
Layer containing class breaks	`INPUT_TABLE`	[vector: any]	Vector layer containing the values to use for classification.
Minimum class value field	`MIN_FIELD`	[lentelės laukas: skaičius]	Field with the minimum value of the range for the class. Use `-inf` to capture the lowest possible value.
Maximum class value field	`MAX_FIELD`	[lentelės laukas: skaičius]	Field with the maximum value of the range for the class. Use `inf` to capture the highest possible value.
Output value field	`VALUE_FIELD`	[lentelės laukas: skaičius]	Field with the value that will be assigned to the pixels that fall in the class (between the corresponding min and max values). Use `nan` to set the value of the range to NoData.
Reclassified raster	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties NėraDuomenų reikšmė	`NO_DATA`	[skaičius: slankaus kablelio] Numatyta: -9999.0	Value to apply to NoData values.
Range boundaries	`RANGE_BOUNDARI`	[sąrašas] Numatytas: 0	Defines comparison rules for the classification. Options: • 0 — min < value <= max • 1 — min <= value < max • 2 — min <= value <= max • 3 — min < value < max
Use NoData when no range matches value	`NODATA_FOR_MIS`	[boolean] Numatytas: ne	Applies the NoData value to band values that do not fall in any class. If False, the original value is kept.

continues on next page

Table 24.374 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas:	Apibrėžia išvesties rastro failo formatą. Parinktys: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — Int32 (Thirty two bit signed integer (qint32)) • 4 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Reclassified raster	`OUTPUT`	[rastras]	Output raster layer with reclassified band values

Pythono kodas

Algorithm ID: `native:reclassifybylayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Reclassify by table

Reclassifies a raster band by assigning new class values based on the ranges specified in a fixed table.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Rastro sluoksnis	`INPUT_RASTER`	[rastras]	Raster layer to reclassify
Juostos numeris	`RASTER_BAND`	[rastro juosta] Numatytas: 1	Raster band for which you want to recalculate values.
Reclassification table	`TABLE`	[vektorius: lentelė]	A 3-columns table to fill with the values to set the boundaries of each class (`Minimum` and `Maximum`) and the new `Value` to assign to the band values that fall in the class. The value `-inf` can be used as minimum, `inf` as maximum, and `nan` can be used to set the output value to NoData.
Reclassified raster	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	I6vesties rastro sluoksnio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties NėraDuomenų reikšmė	`NO_DATA`	[skaičius: slankaus kablelio] Numatyta: -9999.0	Value to apply to NoData values.
Range boundaries	`RANGE_BOUNDARI`	[sąrašas] Numatytas: 0	Defines comparison rules for the classification. Options: • 0 — min < value <= max • 1 — min <= value < max • 2 — min <= value <= max • 3 — min < value < max

continues on next page

Table 24.377 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Use NoData when no range matches value	NODATA_FOR_MIS	[boolean] Numatytas: ne	Applies the NoData value to band values that do not fall in any class. If False, the original value is kept.
Išvesties duomenų tipas	DATA_TYPE	[sąrašas] Numatytas:	Apibrėžia išvesties rastro failo formatą. Parinktys: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — Int32 (Thirty two bit signed integer (qint32)) • 4 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)
Kūrimo parinktys Pasirinktinis	CREATION_OPTIO (for QGIS <= 3.42, this was CREATE_OPTIONS	[tekstas] Numatytas: ,‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Reclassified raster	OUTPUT	[rastras]	Output raster layer with reclassified band values

Pythono kodas

Algorithm ID: `native:reclassifybytable`

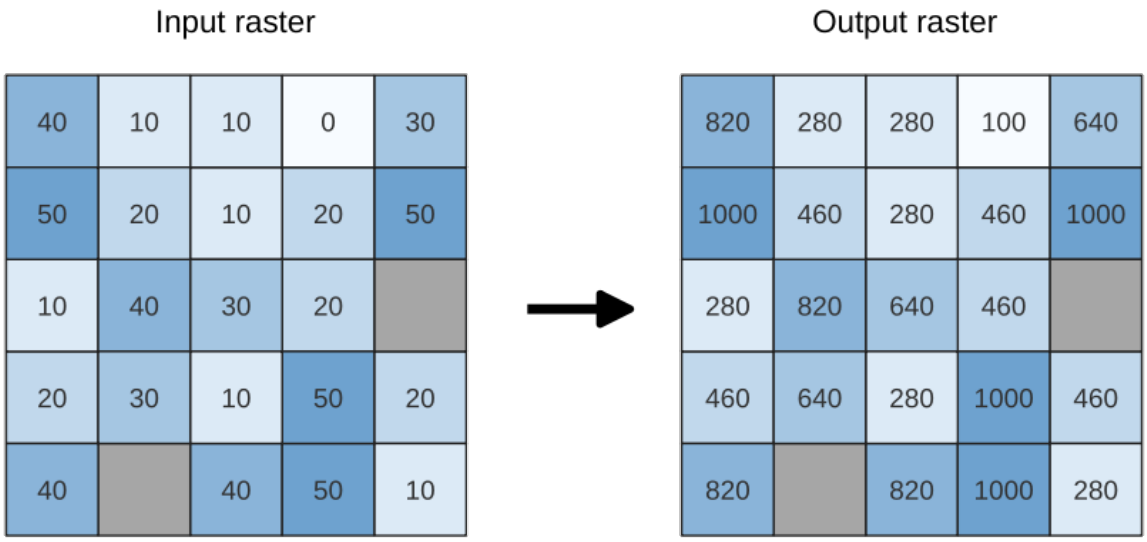
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Rescale raster

Rescales raster layer to a new value range, while preserving the shape (distribution) of the raster's histogram (pixel values). Input values are mapped using a linear interpolation from the source raster's minimum and maximum pixel values to the destination minimum and miximum pixel range.

By default the algorithm preserves the original NoData value, but there is an option to override it.



24.57 Fig. : Rescaling values of a raster layer from [0 - 50] to [100 - 1000]

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input Raster	`INPUT`	[rastras]	Raster layer to use for rescaling
Juostos numeris	`BAND`	[rastro juosta] Numatytas: pirma įvesties sluoksnio juosta	If the raster is multiband, choose a band.
New minimum value	`MINIMUM`	[skaičius: slankaus kablelio] Default value: 0.0	Minimum pixel value to use in the rescaled layer
New maximum value	`MAXIMUM`	[skaičius: slankaus kablelio] Default value: 255.0	Maximum pixel value to use in the rescaled layer
New NoData value Pasirinktinis	`NODATA`	[skaičius: slankaus kablelio] Default value: Not set	Value to assign to the NoData pixels. If unset, original NoData values are preserved.
Rescaled	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	I6vesties rastro sluoksnio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Rescaled	`OUTPUT`	[rastras]	Output raster layer with rescaled band values

Pythono kodas

Algorithm ID: `native:rescaleraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

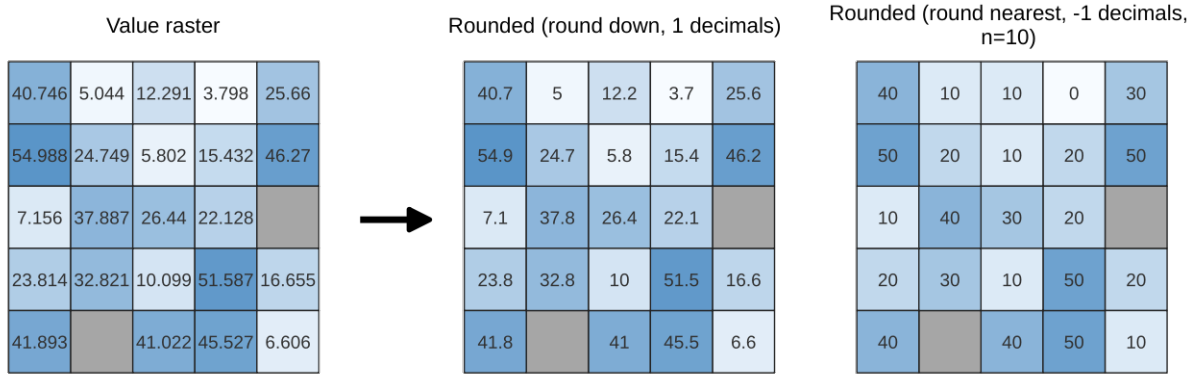
algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Round raster

Rounds the cell values of a raster dataset according to the specified number of decimals.

Alternatively, a negative number of decimal places may be used to round values to powers of a base n. For example, with a Base value n of 10 and Decimal places of -1, the algorithm rounds cell values to multiples of 10, -2 rounds to multiples of 100, and so on. Arbitrary base values may be chosen, the algorithm applies the same multiplicative principle. Rounding cell values to multiples of a base n may be used to generalize raster layers.

The algorithm preserves the data type of the input raster. Therefore byte/integer rasters can only be rounded to multiples of a base n, otherwise a warning is raised and the raster gets copied as byte/integer raster.



24.58 Fig. : Rounding values of a raster

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input raster	`INPUT`	[rastras]	The raster to process.
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	The band of the raster
Rounding direction	`ROUNDING_DIREC`	[sąrašas] Numatytas: 1	How to choose the target rounded value. Options are: • 0 — Round up • 1 — Round to nearest • 2 — Round down

continues on next page

Table 24.381 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Number of decimals places	`DECIMAL_PLACES`	[skaičius: sveikas skaičius] Numatyta: 2	Number of decimals places to round to. Use negative values to round cell values to a multiple of a base n
Išvesties rastras	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties failo specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Base n for rounding to multiples of n	`BASE_N`	[skaičius: sveikas skaičius] Numatytas: 10	When the `DECIMAL_PLACES` parameter is negative, raster values are rounded to multiples of the base n value
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (`|`).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastras	`OUTPUT`	[rastras]	The output raster layer with values rounded for the selected band.

Pythono kodas

Algorithm ID: `native:roundrastervalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Sample raster values

Extracts raster values at the point locations. If the raster layer is multiband, each band is sampled.

The attribute table of the resulting layer will have as many new columns as the raster layer band count.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vector: point]	Point vector layer to use for sampling
Raster Layer	`RASTERCOPY`	[rastras]	Raster layer to sample at the given point locations.
Output column prefix	`COLUMN_PREFIX`	[tekstas] Default: ‚SAMPLE_‘	Prefix for the names of the added columns.
Sampled Pasirinktinis	`OUTPUT`	[vector: point] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output layer containing the sampled values. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sampled	`OUTPUT`	[vector: point]	The output layer containing the sampled values.

Pythono kodas

Algorithm ID: `native:rastersampling`

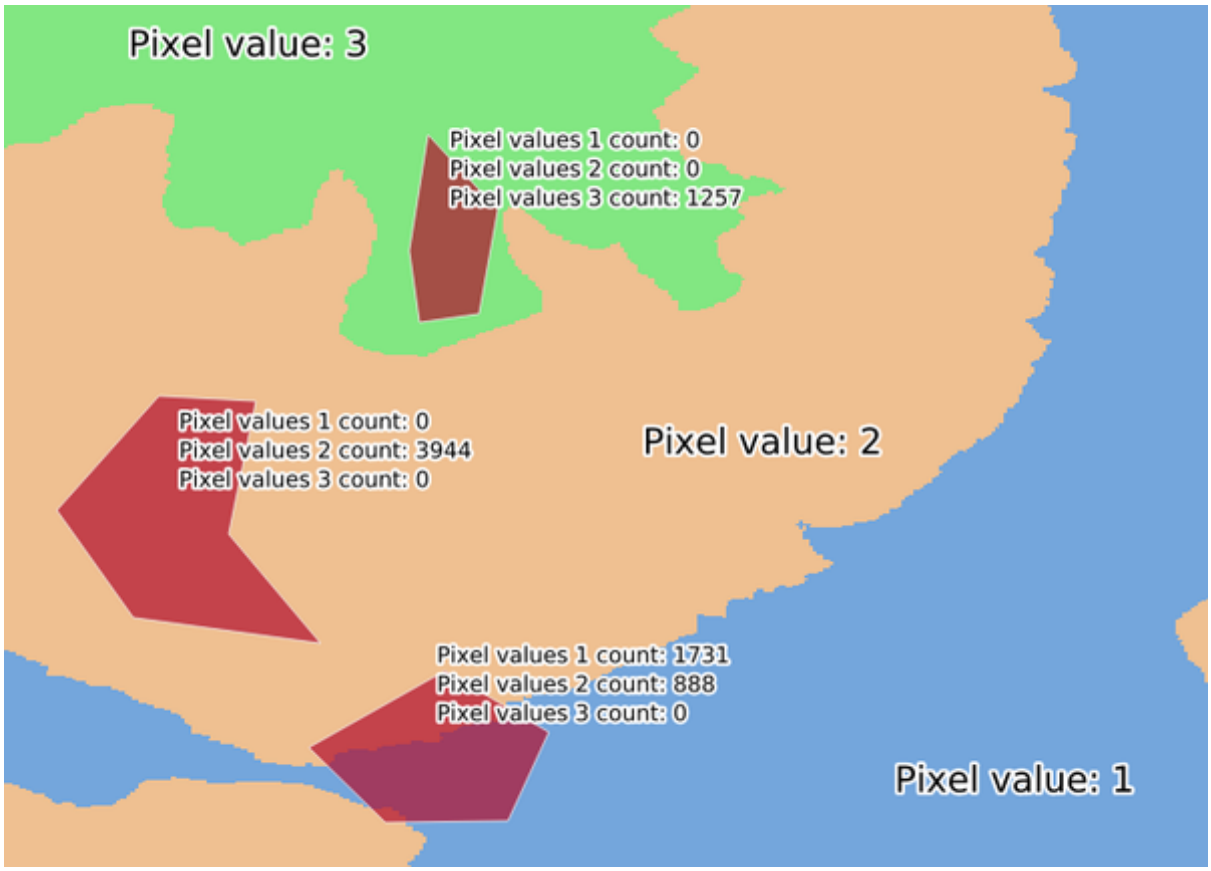
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Zonal histogram

Appends fields representing counts of each unique value from a raster layer contained within polygon features.

The output layer attribute table will have as many fields as the unique values of the raster layer that intersects the polygon(s).



24.59 Fig. : Raster layer histogram example

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Rastro sluoksnis	`INPUT_RASTER`	[rastras]	Input raster layer.
Juostos numeris	`RASTER_BAND`	[rastro juosta] Numatytas: pirma įvesties sluoksnio juosta	If the raster is multiband, choose a band.
Vector layer containing zones	`INPUT_VECTOR`	[vektorius: poligonas]	Vector polygon layer that defines the zones.
Output column prefix	`COLUMN_PREFIX` Pasirinktinis	[tekstas] Default: ‚HISTO_‘	Prefix for the output columns names.

continues on next page

Table 24.385 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Output zones	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector polygon layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Output zones	`OUTPUT`	[vektorius: poligonas]	The output vector polygon layer.

Pythono kodas

Algorithm ID: `native:zonalhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Zonal Minimum/Maximum Point

Extracts point features corresponding to the minimum and maximum pixel values within polygon zones.

The output will contain one point feature for the minimum and one for the maximum raster value for every individual zonal feature from a polygon layer. The created point layer will be in the same spatial reference system as the selected raster layer.

Įspėjimas

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Polygon layer defining the zones.
Raster Layer	`RASTER`	[rastras]	Raster layer to extract the minimum and maximum values from.
Raster Band	`BAND`	[rastro juosta] Default: The first band of the raster layer	If the raster has multiple bands, select the band to process.
Zonal extrema	`OUTPUT`	[vector: point] Numatytas: `[Kurti laikiną sluoksnį]`	Specification of the output layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Zonal extrema	`OUTPUT`	[vector: point]	Output layer containing the minimum and maximum points for each zone.

Pythono kodas

Algorithm ID: `native:zonalminmaxpoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Zonal statistics

Calculates statistics of a raster layer for each feature of an overlapping polygon vector layer.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Vector polygon layer that contains the zones.
Rastro sluoksnis	`INPUT_RASTER`	[rastras]	Input raster layer.
Raster band	`RASTER_BAND`	[rastro juosta] Numatytas: pirma įvesties sluoksnio juosta	If the raster is multiband, choose a band for the statistics.
Output column prefix	`COLUMN_PREFIX`	[tekstas] Default: ‚_‘	Prefix for the output columns names.
Statistics to calculate	`STATISTICS`	[enumeration] [list] Default: [0,1,2]	List of statistical operator for the output. Options: • 0 — Count • 1 — Sum • 2 — Mean • 3 — Median • 4 — St. dev. • 5 — Minimum • 6 — Maximum • 7 — Range • 8 — Minority • 9 — Majority • 10 — Variety • 11 — Variance
Zonal Statistics	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector polygon layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Zonal Statistics	`OUTPUT`	[vektorius: poligonas]	The zone vector layer with added statistics.

Pythono kodas

Algorithm ID: `native:zonalstatisticsfb`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.19 Rastro kūrimas

Kurti konstantų rastro sluoksnį

Sukuria nurodytos apimties ir celės dydžio rastro sluoksnį, užpildytą nurodyta reikšme.

Papildomai galima nurodyti išvesties duomenų tipą. Algoritmas nutrauks darbą, jei įvesta reikšmė negali būti išreikšta parinktu išvesties rastro duomenų tipu.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Norima apimtis	`EXTENT`	[apimtis]	Nurodykite išvesties rastro sluoksnio erdvinę apimtį. Ji viduje bus išplėsta iki kaladėlės dydžio kartotinio. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Paskirties CRS	`TARGET_CRS`	[crs] Numatyta: projekto CRS	Išvesties rastro sluoksnio CRS
Taško dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Taško dydis (X=Y) žemėlapio vienetais.

continues on next page

Table 24.389 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Konstantos reikšmė	`NUMBER`	[skaičius: slankaus kablelio] Numatytas: 1.0	Nekintanti taško reikšmė išvesties rastro sluoksniui.
Konstanta	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastro duomenų tipas	`OUTPUT_TYPE` Numatytas: 5	[sąrašas]	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Byte • 1 — Integer16 • 2 — Unsigned Integer16 • 3 — Integer32 • 4 — Unsigned Integer32 • 5 — Float32 • 6 — Float64
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Konstanta	`OUTPUT`	[rastras]	Nurodyto taško dydžio ir reikšmės rastras, dengiantis norimą apimtį.

Pythono kodas

Algoritmo ID: `native:createconstantrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Kurti atsitiktinį rastro sluoksnį (binominis pasiskirstymas)

Sukuria nurodytos apimties ir celės dydžio rastro sluoksnį užpildytą binomiškai pasiskirsčiusiomis atsitiktinėmis reikšmėmis.

Pagal nutylėjimą reikšmės parenkamus su N=10 ir tikimybe 0,5. Tai galima pakeisti naudojant sudėtingesnį parametrą, nurodant N ir tikimybę. Rastro duomenų tipas yra sveikų skaičių (pagal nutylėjimą Integer16). Binominio pasiskirstymo atsitiktinės reikšmės apibrėžiamos kaip teigiami sveiki skaičiai. Slankaus kablelio rastras reprezentuos sveikus skaičius pakeistus į slankaus kablelio skaičius.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Norima apimtis	`EXTENT`	[apimtis]	Nurodykite išvesties rastro sluoksnio erdvinę apimtį. Ji viduje bus išplėsta iki kaladėlės dydžio kartotinio. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Paskirties CRS	`TARGET_CRS`	[crs] Numatyta: projekto CRS	Išvesties rastro sluoksnio CRS
Taško dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Taško dydis (X=Y) žemėlapio vienetais.

continues on next page

Table 24.392 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastras	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastro duomenų tipas	`OUTPUT_TYPE` Numatytas: 0	[sąrašas]	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Integer16 • 1 — Unsigned Integer16 • 2 — Integer32 • 3 — Unsigned Integer32 • 4 — Float32 • 5 — Float64
N	`N`	[skaičius: sveikas skaičius] Numatytas: 10	
Tikimybė	`PROBABILITY`	[skaičius: slankaus kablelio] Numatyta: 0.5	
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastras	`OUTPUT`	[rastras]	Nurodyto taško dydžio ir dengiantis norimą apimtį rastras, užpildytas atsitiktinėmis reikšmėmis.

Pythono kodas

Algoritmo ID: `native:createrandombinomialrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Kurti atsitiktinį rastro sluoksnį (eksponentinis pasiskirstymas)

Sukuria nurodytos apimties ir celės dydžio rastro sluoksnį užpildytą eksponentiškai pasiskirsčiusiomis atsitiktinėmis reikšmėmis.

Pagal nutylėjimą reikšmės bus parenkamos pagal duotą lambdą 1.0. Galima tai permušti naudojant sudėtingesnį lambdos parametrą. Rastro duomenų tipas pagal nutylėjimą Float32, nes eksponentinio pasiskirstymo atsitiktinės reikšmės yra slankaus kablelio skaičiai.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Norima apimtis	`EXTENT`	[apimtis]	Nurodykite išvesties rastro sluoksnio erdvinę apimtį. Ji viduje bus išplėsta iki kaladėlės dydžio kartotinio. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Paskirties CRS	`TARGET_CRS`	[crs] Numatyta: projekto CRS	Išvesties rastro sluoksnio CRS
Taško dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Taško dydis (X=Y) žemėlapio vienetais.

continues on next page

Table 24.395 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastras	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastro duomenų tipas	`OUTPUT_TYPE` Numatytas: 0	[sąrašas]	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Float32 • 1 — Float64
Lambda	`LAMBDA`	[skaičius: slankaus kablelio] Numatytas: 1.0	
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastras	`OUTPUT`	[rastras]	Nurodyto taško dydžio ir dengiantis norimą apimtį rastras, užpildytas atsitiktinėmis reikšmėmis.

Pythono kodas

Algoritmo ID: `native:createrandomexponentialrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Create random raster layer (gamma distribution)

Generates a raster layer for given extent and cell size filled with gamma distributed random values.

By default, the values will be chosen given an alpha and beta value of 1.0. This can be overridden by using the advanced parameter for alpha and beta. The raster data type is set to Float32 by default as the gamma distribution random values are floating point numbers.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Norima apimtis	`EXTENT`	[apimtis]	Nurodykite išvesties rastro sluoksnio erdvinę apimtį. Ji viduje bus išplėsta iki kaladėlės dydžio kartotinio. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Paskirties CRS	`TARGET_CRS`	[crs] Numatyta: projekto CRS	Išvesties rastro sluoksnio CRS
Taško dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Taško dydis (X=Y) žemėlapio vienetais.
Išvesties rastras	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastro duomenų tipas	`OUTPUT_TYPE` Numatytas: 0	[sąrašas]	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Float32 • 1 — Float64
Alpha	`ALPHA`	[skaičius: slankaus kablelio] Numatytas: 1.0	
Beta	`BETA`	[skaičius: slankaus kablelio] Numatytas: 1.0	
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastras	`OUTPUT`	[rastras]	Raster covering the desired extent with the cell size filled with randomly distributed values

Pythono kodas

Algorithm ID: `native:createrandomgammarasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Create random raster layer (geometric distribution)

Generates a raster layer for given extent and cell size filled with geometrically distributed random values.

By default, the values will be chosen given a probability of 0.5. This can be overridden by using the advanced parameter for mean value. The raster data type is set to Integer types (Integer16 by default). The geometric distribution random values are defined as positive integer numbers. A floating point raster will represent a cast of integer values to floating point.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Norima apimtis	`EXTENT`	[apimtis]	Nurodykite išvesties rastro sluoksnio erdvinę apimtį. Ji viduje bus išplėsta iki kaladėlės dydžio kartotinio. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Paskirties CRS	`TARGET_CRS`	[crs] Numatyta: projekto CRS	Išvesties rastro sluoksnio CRS
Taško dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Taško dydis (X=Y) žemėlapio vienetais.
Išvesties rastras	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastro duomenų tipas	`OUTPUT_TYPE` Numatytas: 0	[sąrašas]	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Integer16 • 1 — Unsigned Integer16 • 2 — Integer32 • 3 — Unsigned Integer32 • 4 — Float32 • 5 — Float64
Tikimybė	`PROBABILITY`	[skaičius: slankaus kablelio] Numatyta: 0.5	
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastras	`OUTPUT`	[rastras]	Raster covering the desired extent with the cell size filled with randomly distributed values

Pythono kodas

Algorithm ID: `native:createrandomgeometricrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Create random raster layer (negative binomial distribution)

Generates a raster layer for given extent and cell size filled with negative binomially distributed random values.

By default, the values will be chosen given a distribution parameter k of 10.0 and a probability of 0.5. This can be overridden by using the advanced parameters for k and probability. The raster data type is set to Integer types (Integer16 by default). The negative binomial distribution random values are defined as positive integer numbers. A floating point raster will represent a cast of integer values to floating point.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Norima apimtis	`EXTENT`	[apimtis]	Nurodykite išvesties rastro sluoksnio erdvinę apimtį. Ji viduje bus išplėsta iki kaladėlės dydžio kartotinio. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Paskirties CRS	`TARGET_CRS`	[crs] Numatyta: projekto CRS	Išvesties rastro sluoksnio CRS
Taško dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Taško dydis (X=Y) žemėlapio vienetais.
Išvesties rastras	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastro duomenų tipas	`OUTPUT_TYPE` Numatytas: 0	[sąrašas]	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Integer16 • 1 — Unsigned Integer16 • 2 — Integer32 • 3 — Unsigned Integer32 • 4 — Float32 • 5 — Float64
Distribution parameter k	`K_PARAMETER`	[skaičius: sveikas skaičius] Numatytas: 10	
Tikimybė	`PROBABILITY`	[skaičius: slankaus kablelio] Numatyta: 0.5	
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastras	`OUTPUT`	[rastras]	Raster covering the desired extent with the cell size filled with randomly distributed values

Pythono kodas

Algorithm ID: `native:createrandomnegativebinomialrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Create random raster layer (normal distribution)

Generates a raster layer for given extent and cell size filled with normally distributed random values.

By default, the values will be chosen given a mean of 0.0 and a standard deviation of 1.0. This can be overridden by using the advanced parameters for mean and standard deviation value. The raster data type is set to Float32 by default as the normal distribution random values are floating point numbers.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Norima apimtis	`EXTENT`	[apimtis]	Nurodykite išvesties rastro sluoksnio erdvinę apimtį. Ji viduje bus išplėsta iki kaladėlės dydžio kartotinio. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Paskirties CRS	`TARGET_CRS`	[crs] Numatyta: projekto CRS	Išvesties rastro sluoksnio CRS
Taško dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Taško dydis (X=Y) žemėlapio vienetais.
Išvesties rastras	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastro duomenų tipas	`OUTPUT_TYPE` Numatytas: 0	[sąrašas]	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Float32 • 1 — Float64
Mean of normal distribution	`MEAN`	[skaičius: slankaus kablelio] Numatytas: 0.0	
Standard deviation of normal distribution	`STDDEV`	[skaičius: slankaus kablelio] Numatytas: 1.0	
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastras	`OUTPUT`	[rastras]	Raster covering the desired extent with the cell size filled with randomly distributed values

Pythono kodas

Algorithm ID: `native:createrandomnormalrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Create random raster layer (poisson distribution)

Generates a raster layer for given extent and cell size filled with poisson distributed random values.

By default, the values will be chosen given a mean of 1.0. This can be overridden by using the advanced parameter for mean value. The raster data type is set to Integer types (Integer16 by default). The poisson distribution random values are positive integer numbers. A floating point raster will represent a cast of integer values to floating point.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Norima apimtis	`EXTENT`	[apimtis]	Nurodykite išvesties rastro sluoksnio erdvinę apimtį. Ji viduje bus išplėsta iki kaladėlės dydžio kartotinio. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Paskirties CRS	`TARGET_CRS`	[crs] Numatyta: projekto CRS	Išvesties rastro sluoksnio CRS
Taško dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Taško dydis (X=Y) žemėlapio vienetais.
Išvesties rastras	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastro duomenų tipas	`OUTPUT_TYPE` Numatytas: 0	[sąrašas]	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Integer16 • 1 — Unsigned Integer16 • 2 — Integer32 • 3 — Unsigned Integer32 • 4 — Float32 • 5 — Float64
Mean	`MEAN`	[skaičius: slankaus kablelio] Numatytas: 1.0	
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastras	`OUTPUT`	[rastras]	Raster covering the desired extent with the cell size filled with randomly distributed values

Pythono kodas

Algorithm ID: `native:createrandompoissonrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Create random raster layer (uniform distribution)

Generates a raster layer for given extent and cell size filled with random values.

By default, the values will range between the minimum and maximum value of the specified output raster type. This can be overridden by using the advanced parameters for lower and upper bound value. If the bounds have the same value or both are zero (default) the algorithm will create random values in the full value range of the chosen raster data type. Choosing bounds outside the acceptable range of the output raster type will abort the algorithm.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Norima apimtis	`EXTENT`	[apimtis]	Nurodykite išvesties rastro sluoksnio erdvinę apimtį. Ji viduje bus išplėsta iki kaladėlės dydžio kartotinio. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Paskirties CRS	`TARGET_CRS`	[crs] Numatyta: projekto CRS	Išvesties rastro sluoksnio CRS
Taško dydis	`PIXEL_SIZE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Taško dydis (X=Y) žemėlapio vienetais.
Išvesties rastras	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastro duomenų tipas	`OUTPUT_TYPE` Numatytas: 5	[sąrašas]	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Byte • 1 — Integer16 • 2 — Unsigned Integer16 • 3 — Integer32 • 4 — Unsigned Integer32 • 5 — Float32 • 6 — Float64
Lower bound for random number range	`LOWER_BOUND`	[skaičius: slankaus kablelio] Numatytas: 0.0	
Upper bound for random number range	`UPPER_BOUND`	[skaičius: slankaus kablelio] Numatytas: 0.0	
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastras	`OUTPUT`	[rastras]	Raster covering the desired extent with the cell size filled with randomly distributed values

Pythono kodas

Algorithm ID: `native:createrandomuniformrasterlayer`

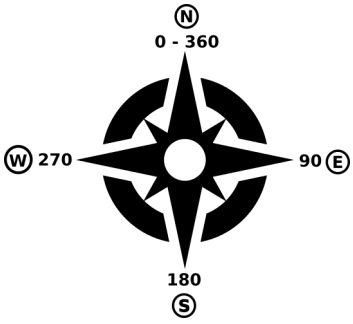
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.20 Rastro paviršiaus analizė

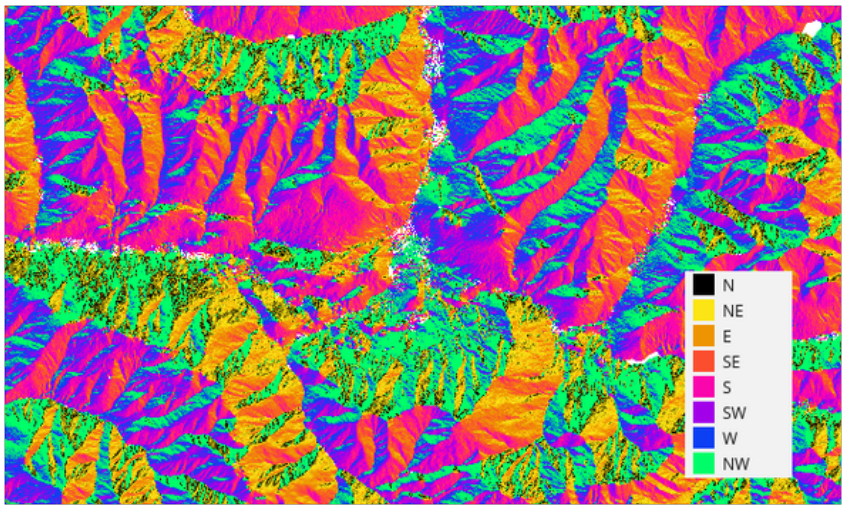
Aspektas

Skaičiuoja įvestyje pateikto skaitmeninio paviršiaus modelio aspektą. Gautame aspekto rastro sluoksnyje yra reikšmės nuo 0 iki 360, kurios išreiškia nuolydžio kryptį, pradedant nuo šiaurės (0°) ir skaičiuojant prieš laikrodžio rodyklę.



24.60 Fig. : Aspekto reikšmės

Žemiau pateikiama iliustracija, kur aspekto sluoksnis perklasifikuotas naudojant spalvų rampą:



24.61 Fig. : Perklasifikuotas aspekto sluoksnis

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Aukščio sluoksnis	`INPUT`	[rastras]	Skaitmeninio paviršiaus modelio rastro sluoksnis
Z koeficientas	`Z_FACTOR`	[skaičius: slankaus kablelio] Numatytas: 1.0	Vertikalus didinimas. Šis parametras naudingas, kai Z vienetai skiriasi nuo X ir Y vienetų, pavyzdžiui pėdos ir metrai. Jūs galite naudoti šį parametrą, kad prisiderintumėte prie šito. Numatytoji reikšmė yra 1 (be padidinimo).

continues on next page

Table 24.416 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Aspektas	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite išvesties aspekto rastro sluoksnį. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Aspektas	`OUTPUT`	[rastras]	Išvesties aspekto rastro sluoksnis

Pythono kodas

Algorithm ID: `native:aspect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

DTM filtras (pagal nuolydį)

Gali būti naudojamas filtruojant skaitmeninį aukščio modelį klasifikuojant jo celes į žemės ir objektų (ne žemės) celes.

Įrankis naudoja koncepcijas, apibrėžtas Vosselman (2000) ir remiasi prielaida, kad dideli aukščio skirtumai tarp gretimų celių negali būti dėl paviršiaus nuolydžio. Tikimybė, kad aukštesnės celės gali būti ne žemė didėja mažėjant atstumui tarp dviejų celių. Todėl filtras apibrėžia maksimalų aukščio skirtumą (`dz_max`) tarp celių kaip funkciją atstumo (`d`) tarp celių (`dz_max( d ) = d`). Celė klasifikuojama kaip paviršiaus, jei branduolio spindulio atstumu nėra celės, kurios aukščio skirtumas būtų didesnis nei nurodytas aukščio skirtumų maksimumas nurodytu atstumu tarp dviejų celių.

Apytikslio paviršiaus nuolydžio (`s`) parametras naudojamas keisti filtro funkciją, kad ji atitiktų bendrą nuolydį studijuojamame plote (`dz_max( d ) = d * s`). 5 % užtikrintumo intervalas (`ci = 1.65 * sqrt( 2 * stddev )`) gali būti naudojamas keisti filtro funkciją dar labiau, arba atlaisvinant (`dz_max( d ) = d * s + ci`), arba padidinant (`dz_max( d ) = d * s - ci`) filtro kriterijų.

Šaltinis: Vosselman, G. (2000): Slope based filtering of laser altimetry data. IAPRS, Vol. XXXIII, Part B3, Amsterdam, The Netherlands, 935-942

Taip pat žiūrėkite

This tool is a port of the SAGA DTM Filter (slope-based)

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Skaitmeninio paviršiaus modelio rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta]	Tiriama DEM juosta
Branduolio spindulys (pikseliais)	`RADIUS`	[skaičius: sveikas skaičius] Numatytas:	Filtro branduolio spindulys (pikseliais). Turi būti pakankamai didelis, kad pasiektų paviršiaus celes greta ne žemės objektų.
Paviršiaus nuolydis (%, pikselio dydis/vertikalūs vienetai)	`TERRAIN_SLOPE`	[skaičius: slankaus kablelio] Numatytas: 30.0	Apytikslis paviršiaus nuolydis `%`. Paviršiaus nuolydį reikia patikslinti, kad būtų atsižvelgta į aukščio vienetų ir rastro pikselių matmenų santykį. Naudojama atlaisvinti filtro kriterijų statesniame paviršiuje.
Filtro modifikacija	`FILTER_MODIFIC`	[sąrašas] Numatytas: 0	Pasirinkite, ar taikyti filtro branduolį be pokyčių, ar naudoti užtikrintumo intervalą aukščio kriterijaus atlaisvinimui ar padidinimui. • 0 - Jokio • 1 - Atlaisvinti filtrą • 2 - Padidinti
Standartinis nuokrypis	`STANDARD_DEVIA`	[skaičius: slankaus kablelio] Numatytas: 0.1	Standartinis nuokrypis naudojamas skaičiuojant 5% užtikrintumo intervalą, taikomą aukščio slenkščiui.
Išvesties sluoksnis (žemė) Pasirinktinis	`OUTPUT_GROUND`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite filtruotą DEM, kuriame yra tik celės, klasifikuotos kaip žemė. *Vienas iš*: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…
Išvesties sluoksnis (ne žemės objektai) Pasirinktinis	`OUTPUT_NONGROU`	[rastras] Numatytas: `[Praleisti išvestį]`	Nurodykite ne žemės objektus, pašalintus filtro. *Vienas iš*: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis (žemė)	`OUTPUT_GROUND`	[rastras]	Filtruotas DEM, kuriame yra tik celės, klasifikuotos kaip žemė.
Išvesties sluoksnis (ne žemės objektai)	`OUTPUT_NONGROU`	[rastras]	Ne žemės objektai, kuriuos pašalino filtras.

Pythono kodas

Algoritmo ID: `native:dtmslopebasedfilter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Fill sinks (Wang & Liu)

`NEW in 3.44`

Uses a method proposed by Wang & Liu to identify and fill surface depressions in digital elevation models.

The method was enhanced to allow the creation of hydrologically sound elevation models, i.e. not only to fill the depression(s) but also to preserve a downward slope along the flow path. If desired, this is accomplished by preserving a minimum slope gradient (and thus elevation difference) between cells.

References: Wang, L. & H. Liu (2006): An efficient method for identifying and filling surface depressions in digital elevation models for hydrologic analysis and modelling. International Journal of Geographical Information Science, Vol. 20, No. 2: 193-213.

Taip pat žiūrėkite

This tool is a port of the SAGA Fill Sinks (Wang & Liu) tool.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Skaitmeninio paviršiaus modelio rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta] Default: 1	Tiriama DEM juosta
Minimum slope (degrees)	`MIN_SLOPE`	[skaičius: slankaus kablelio] Default: 0.10	Minimum slope gradient to preserve from cell to cell; with a value of zero, sinks are filled up to the spill elevation (which results in flat areas).
Output layer (filled DEM) Pasirinktinis	`OUTPUT_FILLED_`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the ouput raster corresponding to the depression-free DEM. *One of*: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…
Output layer (flow directions) Pasirinktinis	`OUTPUT_FLOW_DI`	[rastras] Numatytas: `[Praleisti išvestį]`	Specify the output raster with computed flow directions. *One of*: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…
Output layer (watershed basins) Pasirinktinis	`OUTPUT_WATERSH`	[rastras] Numatytas: `[Praleisti išvestį]`	Specify the output raster corresponding to the delineated watershed basins. *One of*: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Output layer (filled DEM)	`OUTPUT_FILLED_`	[rastras]	Ouput raster corresponding to the depression-free digital elevation model.
Output layer (flow directions)	`OUTPUT_FLOW_DI`	[rastras]	Output raster with computed flow directions; 0=N, 1=NE, 2=E, … 7=NW.
Output layer (watershed basins)	`OUTPUT_WATERSH`	[rastras]	Output raster corresponding to the delineated watershed basins.

Pythono kodas

Algorithm ID: `native:fillsinkswangliu`

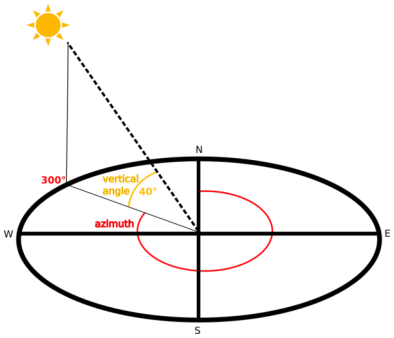
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Hillshade

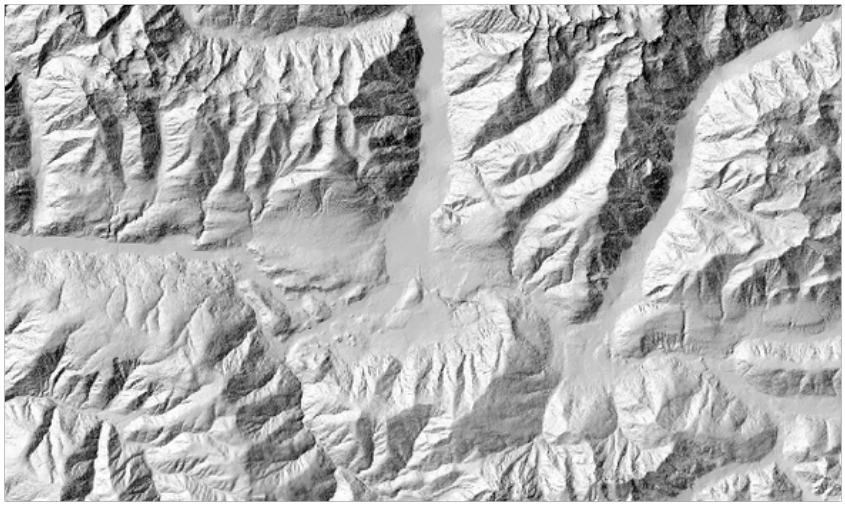
Calculates the hillshade raster layer given an input Digital Terrain Model.

The shading of the layer is calculated according to the sun position: you have the options to change both the horizontal angle (azimuth) and the vertical angle (sun elevation) of the sun.



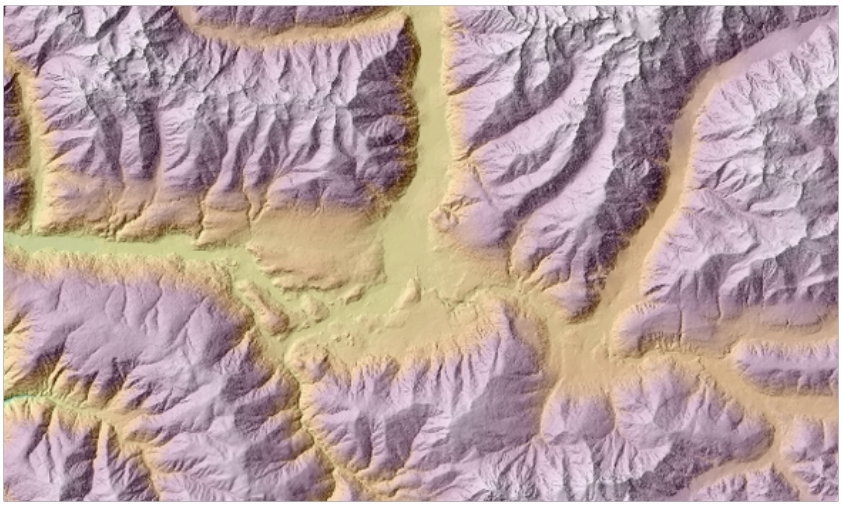
24.62 Fig. : Azimuth and vertical angle

The hillshade layer contains values from 0 (complete shadow) to 255 (complete sun). Hillshade is used usually to better understand the relief of the area.



24.63 Fig. : Hillshade layer with azimuth 300 and vertical angle 45

Particularly interesting is to give the hillshade layer a transparency value and overlap it with the elevation raster:



24.64 Fig. : Overlapping the hillshade with the elevation layer

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Aukščio sluoksnis	`INPUT`	[rastras]	Skaitmeninio paviršiaus modelio rastro sluoksnis
Z koeficientas	`Z_FACTOR`	[skaičius: slankaus kablelio] Numatytas: 1.0	Vertical exaggeration. This parameter is useful when the Z units differ from the X and Y units, for example feet and meters. You can use this parameter to adjust for this. Increasing the value of this parameter will exaggerate the final result (making it look more „hilly“). The default is 1 (no exaggeration).

continues on next page

Table 24.424 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Azimuth (horizontal angle)	`AZIMUTH`	[skaičius: slankaus kablelio] Default: 300.0	Set the horizontal angle (in degrees) of the sun (clockwise direction). Range: 0 to 360. 0 is north.
Vertical angle	`V_ANGLE`	[skaičius: slankaus kablelio] Default: 40.0	Set the vertical angle (in degrees) of the sun, that is the height of the sun. Values can go from 0 (minimum elevation) to 90 (maximum elevation).
Hillshade	`OUTPUT`	[rastras] Default: `Save to temporary file`	Specify the output hillshade raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Hillshade	`OUTPUT`	[rastras]	The output hillshade raster layer

Pythono kodas

Algorithm ID: `native:hillshade`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Hypsometric curves

Calculates hypsometric curves for an input Digital Elevation Model. Curves are produced as CSV files in an output folder specified by the user.

A hypsometric curve is a cumulative histogram of elevation values in a geographical area.

You can use hypsometric curves to detect differences in the landscape due to the geomorphology of the territory.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
DEM to analyze	`INPUT_DEM`	[rastras]	Digital Terrain Model raster layer to use for calculating altitudes
Boundary layer	`BOUNDARY_LAYER`	[vektorius: poligonas]	Polygon vector layer with boundaries of areas used to calculate hypsometric curves
Step	`STEP`	[skaičius: slankaus kablelio] Numatytas: 100.0	Vertical distance between curves

continues on next page

Table 24.426 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Use % of area instead of absolute value	`USE_PERCENTAGE`	[boolean] Numatytas: ne	Write area percentage to “Area” field of the CSV file instead of the absolute area
Hypsometric curves	`OUTPUT_DIRECTO`	[aplankas]	Specify the output folder for the hypsometric curves. *One of*: • Įrašyti į laikiną aplanką • Įrašyti į aplanką

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Hypsometric curves	`OUTPUT_DIRECTO`	[aplankas]	Directory containing the files with the hypsometric curves. For each feature from the input vector layer, a CSV file with area and altitude values will be created. The file names start with `histogram_`, followed by layer name and feature ID.

	A	B
1	Area	Elevation
2	177475194.383	307
3	233206029.24	407
4	295553735.793	507
5	394718815.615	607
6	501801102.615	707
7	624399019.792	807
8	828877274.39	907
9	1042693465.68	1007
10	1277373021.81	1107
11	1556443975.41	1207
12	1888617494.27	1307
13	2248520437.31	1407
14	2627916813.17	1507
15	3010880212.04	1607
16	3411087555.34	1707

Pythono kodas

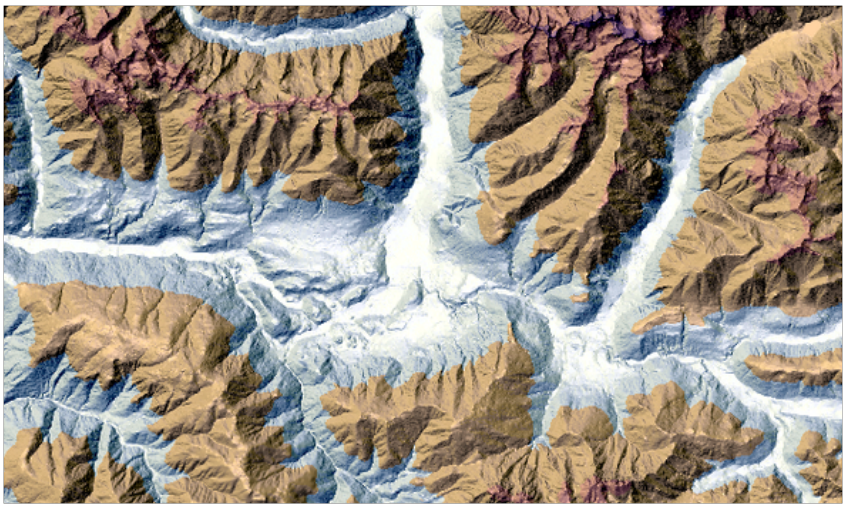
Algorithm ID: `qgis:hypsometriccurves`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Relief

Creates a shaded relief layer from digital elevation data. You can specify the relief color manually, or you can let the algorithm choose automatically all the relief classes.



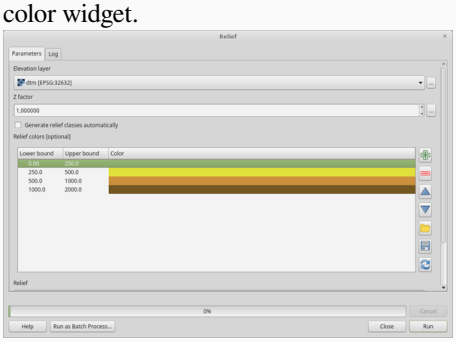
24.65 Fig. : Relief layer

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Aukščio sluoksnis	`INPUT`	[rastras]	Skaitmeninio paviršiaus modelio rastro sluoksnis
Z koeficientas	`Z_FACTOR`	[skaičius: slankaus kablelio] Numatytas: 1.0	Vertical exaggeration. This parameter is useful when the Z units differ from the X and Y units, for example feet and meters. You can use this parameter to adjust for this. Increasing the value of this parameter will exaggerate the final result (making it look more „hilly“). The default is 1 (no exaggeration).
Generate relief classes automatically	`AUTO_COLORS`	[boolean] Numatytas: ne	If you check this option the algorithm will create all the relief color classes automatically

continues on next page

Table 24.428 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Relief colors Pasirinktinis	`COLORS`	[table widget]	Use the table widget if you want to choose the relief colors manually. You can add as many color classes as you want: for each class you can choose the lower and upper bound and finally by clicking on the color row you can choose the color thanks to the color widget.  24.66 Fig. : Manually setting of relief color classes The buttons in the right side panel give you the chance to: add or remove color classes, change the order of the color classes already defined, open an existing file with color classes and save the current classes as file.
Relief	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output relief raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…
Frequency distribution Pasirinktinis	`FREQUENCY_DIST`	[vektorius: lentelė] Numatytas: `[Praleisti išvestį]`	Specify the CSV table for the output frequency distribution. *One of*: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Relief	`OUTPUT`	[rastras]	The output relief raster layer
Frequency distribution	`OUTPUT`	[vektorius: lentelė]	The output frequency distribution

Pythono kodas

Algorithm ID: `qgis:relief`

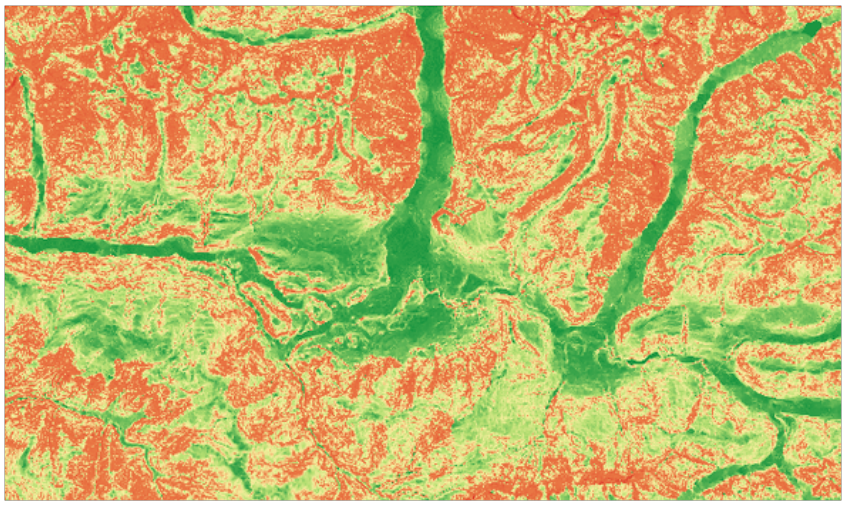
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Ruggedness index

Calculates the quantitative measurement of terrain heterogeneity described by Riley et al. (1999). It is calculated for every location, by summarizing the change in elevation within the 3x3 pixel grid.

Each pixel contains the difference in elevation from a center cell and the 8 cells surrounding it.



24.67 Fig. : Ruggedness layer from low (red) to high values (green)

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Aukščio sluoksnis	`INPUT`	[rastras]	Skaitmeninio paviršiaus modelio rastro sluoksnis
Z koeficientas	`Z_FACTOR`	[skaičius: slankaus kablelio] Numatytas: 1.0	Vertical exaggeration. This parameter is useful when the Z units differ from the X and Y units, for example feet and meters. You can use this parameter to adjust for this. Increasing the value of this parameter will exaggerate the final result (making it look more rugged). The default is 1 (no exaggeration).

continues on next page

Table 24.430 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Ruggedness	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output ruggedness raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Ruggedness	`OUTPUT`	[rastras]	The output ruggedness raster layer

Pythono kodas

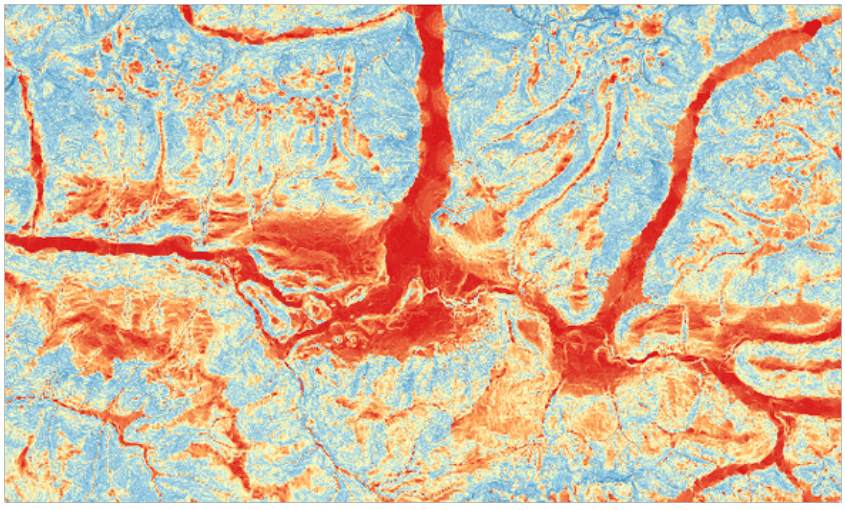
Algorithm ID: `native:ruggednessindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Slope

Calculates the slope from an input raster layer. The slope is the angle of inclination of the terrain and is expressed in **degrees**.



24.68 Fig. : Flat areas in red, steep areas in blue

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Aukščio sluoksnis	`INPUT`	[rastras]	Skaitmeninio paviršiaus modelio rastro sluoksnis
Z koeficientas	`Z_FACTOR`	[skaičius: slankaus kablelio] Numatytas: 1.0	Vertical exaggeration. This parameter is useful when the Z units differ from the X and Y units, for example feet and meters. You can use this parameter to adjust for this. Increasing the value of this parameter will exaggerate the final result (making it steeper). The default is 1 (no exaggeration).
Slope	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output slope raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Slope	`OUTPUT`	[rastras]	The output slope raster layer

Pythono kodas

Algorithm ID: `native:slope`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.21 Rastro įrankiai

Lygiuoti rastrą

Lygiuoja rastrą perrenkant jį į tokį patį celių dydį ir perprojektuojant į tą pačią CRS kaip ir atskaitos rastras.

Įspėjimas

Šis algoritmas veikia TIK *Modelių dizainerio* kontekste. Kituose kontekstuose naudokite *Lygiuoti rastrus*.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Lygiuojamas įvesties rastras
Perrinkimo metodas	`RESAMPLING_MET`	[sąrašas] Numatytas: 0	Method to use for input layer resampling. `Available options` are: • 0 — Artimiausias kaimynas • 1 — Bitiesinis (2x2 branduolio) • 2 — Kubinis (4x4 branduolio) • 3 — Kubinis B-kreivės (4x4 branduolio) • 4 — Lanczos (6x6 branduolio) • 5 — Vidurkis • 6 — Režimas • 7 — Maksimumas • 8 — Minimumas • 9 — Mediana • 10 — Pirmas kvartilis (Q1) • 11 — Trečias kvartilis (Q3)
Perskaičiuoti reikšmes pagal celės dydį	`RESCALE`	[boolean] Numatytas: Ne	
Atskaitos sluoksnis	`REFERENCE_LAYE`	[rastras]	Rastro sluoksnis, kuris bus naudotas ištraukti apimčiai, celių dydžiui ir CRS, kurie bus taikomi įvesties sluoksniams.
Permušti atskaitos CRS Pasirinktinis	`CRS`	[crs]	CRS, kurį naudoti vietoje rasto atskaitos sluoksnyje
Permušti atskaitos celės dydį X Pasirinktinis	`CELL_SIZE_X`	[skaičius: slankaus kablelio]	Celės dydis X kryptimi bus naudojamas vietoje rasto apskaitos sluoksnyje
Permušti atskaitos celės dydį Y Pasirinktinis	`CELL_SIZE_Y`	[skaičius: slankaus kablelio]	Celės dydis Y kryptimi bus naudojamas vietoje rasto apskaitos sluoksnyje
Permušti atskaitos tinklelio poslinkį X Pasirinktinis	`GRID_OFFSET_X`	[skaičius: slankaus kablelio]	Poslinkis X kryptimi, kurį taikyti celių tinkleliui
Permušti atskaitos tinklelio poslinkį Y Pasirinktinis	`GRID_OFFSET_Y`	[skaičius: slankaus kablelio]	Poslinkis Y kryptimi, kurį taikyti celių tinkleliui

continues on next page

Table 24.434 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Apkarpyti pagal apimtį Pasirinktinis	`EXTENT`	[apimtis]	Nurodykite išvesties rastro sluoksnio apimtį. Ji viduje bus išplėsta iki kaladėlės dydžio kartotinio. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Sulygiuotas rastras	`OUTPUT`	[rastras]	išvesties rastras su perrikiuotais taškais

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sulygiuotas rastras	`OUTPUT`	[rastras]	išvesties rastras su perrikiuotais taškais

Pythono kodas

Algoritmo ID: `native:alignsingleraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Lygiuoti rastrus

Lygiuoja rastrus perrenkant juos į tokį patį celių dydį ir perprojektuojant į tą pačią CRS kaip ir atskaitos rastras.

Įspėjimas

Šis algoritmas NEVEIKIA *Modelių dizainerio* kontekste. Vietoje to naudokite *Lygiuoti rastrą*.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`INPUT`	[rastras] [sąrašas]	Sąrašas įvesties rastrų su susijusiomis perrinkimo parinktimis (užpildytomis kaip `QgsProcessingParameterAlignRasterLayers` elementai — atliekama GUI naudojant mygtuką *Konfigūruoti rastrą...* kiekvienam parinktam sluoksniui): **Įvesties sluoksnis [tekstas] (`inputFile`)** Pilnas kelias iki lygiuojamo įvesties sluoksnio **Išvesties failas [tekstas] (`outputFile`)** Pilnas atitinkamai lygiuoto išvesties sluoksnio kelias **Perrinkimo metodas [sąrašas] (`resampleMethod`)** Method to use for input layer resampling. `Available options` are: • 0 — Artimiausias kaimynas • 1 — Bitiesinis (2x2 branduolio) • 2 — Kubinis (4x4 branduolio) • 3 — Kubinis B-kreivės (4x4 branduolio) • 4 — Lanczos (6x6 branduolio) • 5 — Vidurkis • 6 — Režimas • 7 — Maksimumas • 8 — Minimumas • 9 — Mediana • 10 — Pirmas kvartilis (Q1) • 11 — Trečias kvartilis (Q3) **Mastelio keitimo reikšmės pagal celės dydį** [loginis] (`rescale`)
Atskaitos sluoksnis	`REFERENCE_LAYE`	[rastras]	Rastro sluoksnis, kuris bus naudotas ištraukti apimčiai, celių dydžiui ir CRS, kurie bus taikomi įvesties sluoksniams.
Permušti atskaitos CRS Pasirinktinis	`CRS`	[crs]	CRS, kurį naudoti vietoje rasto atskaitos sluoksnyje
Permušti atskaitos celės dydį X Pasirinktinis	`CELL_SIZE_X`	[skaičius: slankaus kablelio]	Celės dydis X kryptimi bus naudojamas vietoje rasto apskaitos sluoksnyje

continues on next page

Table 24.436 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Permušti atskaitos celės dydį Y Pasirinktinis	`CELL_SIZE_Y`	[skaičius: slankaus kablelio]	Celės dydis Y kryptimi bus naudojamas vietoje rasto apskaitos sluoksnyje
Permušti atskaitos tinklelio poslinkį X Pasirinktinis	`GRID_OFFSET_X`	[skaičius: slankaus kablelio]	Poslinkis X kryptimi, kurį taikyti celių tinkleliui
Permušti atskaitos tinklelio poslinkį Y Pasirinktinis	`GRID_OFFSET_Y`	[skaičius: slankaus kablelio]	Poslinkis Y kryptimi, kurį taikyti celių tinkleliui
Apkarpyti pagal apimtį Pasirinktinis	`EXTENT`	[apimtis]	Nurodykite išvesties rastro sluoksnio apimtį. Ji viduje bus išplėsta iki kaladėlės dydžio kartotinio. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Lygiuoti rastrai	`OUTPUT_LAYERS`	[rastras] [sąrašas]	išvesties rastrai su perrinktais pikseliais

Pythono kodas

Algoritmo ID: `native:alignrasters`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Convert map to raster

Creates a raster image of map canvas content.

A *map theme* can be selected to render a predetermined set of layers with a defined style for each layer.

Alternatively, a single layer can be selected if no map theme is set.

If neither map theme nor layer is set, the current map content will be rendered. The minimum extent entered will internally be extended to be a multiple of the tile size.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Minimum extent to render (xmin, xmax, ymin, ymax)	`EXTENT`	[apimtis]	Nurodykite išvesties rastro sluoksnio apimtį. Ji viduje bus išplėsta iki kaladėlės dydžio kartotinio. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Tile size	`TILE_SIZE`	[skaičius: sveikas skaičius] Default: 1024	Size of the tile of the output raster layer. Minimum value: 64.
Map units per pixel	`MAP_UNITS_PER_`	[skaičius: slankaus kablelio] Numatytas: 100.0	Pixel size (in map units). Minimum value: 0.0
Make background transparent	`MAKE_BACKGROUN`	[boolean] Numatytas: ne	Allows exporting the map with a transparent background. Outputs an RGBA (instead of RGB) image if set to `True`.
Map theme to render Pasirinktinis	`MAP_THEME`	[sąrašas]	Use an existing *map theme* for the rendering.
Single layer to render Pasirinktinis	`LAYER`	[sąrašas]	Choose a single layer for the rendering
Išvesties sluoksnis	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	`OUTPUT`	[rastras]	Išvesties rastro sluoksnis

Pythono kodas

Algorithm ID: `native:rasterize`

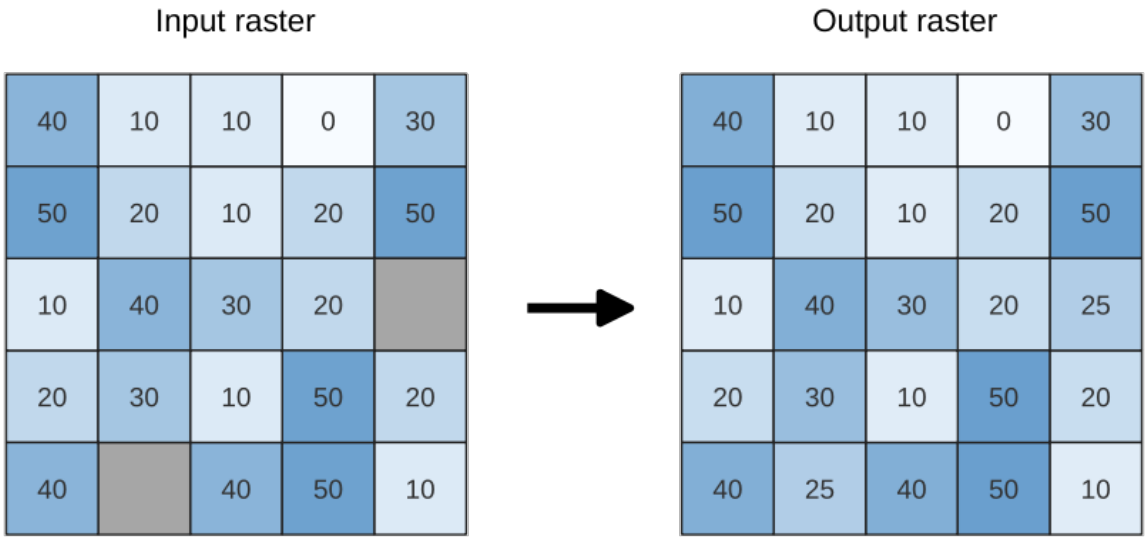
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Fill NoData cells

Resets the NoData values in the input raster to a chosen value, resulting in raster dataset with no NoData pixels.

The algorithm respects the input raster data type, e.g. a floating point fill value will be truncated when applied to an integer raster.



24.69 Fig. : Filling NoData values (in grey) of a raster

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input raster	`INPUT`	[rastras]	The raster to process.
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	The band of the raster
Fill value	`FILL_VALUE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Set the value to use for the NoData pixels
Išvesties rastras	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `CREATE_OPTIONS`	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties rastras	`OUTPUT`	[rastras]	The output raster layer with filled data cells.

Pythono kodas

Algorithm ID: `native:fillnodata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Generate XYZ tiles (Directory)

Generates raster "XYZ" tiles using the current QGIS project as individual images to a directory structure.

Optionally, a Leaflet HTML output file using the generated tiles as a map layer could be created.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Apimtis (xmin, xmax, ymin, ymax)	`EXTENT`	[apimtis]	Specify the extent of the tiles. It will internally be extended to a multiple of the tile size. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Minimum zoom	`ZOOM_MIN`	[skaičius: sveikas skaičius] Default: 12	Minimum 0, maximum 25.
Maximum zoom	`ZOOM_MAX`	[skaičius: sveikas skaičius] Default: 12	Minimum 0, maximum 25.
DPI	`DPI`	[skaičius: sveikas skaičius] Default: 96	Minimum 48, maximum 600.
Background color Pasirinktinis	`BACKGROUND_COL`	[color] Default: QColor(0, 0, 0, 0)	Choose the background color for the tiles
Enable antialiasing	`ANTIALIAS`	[boolean] Numatytas: Taip	Determines if antialiasing should be enabled
Tile format	`TILE_FORMAT`	[sąrašas] Numatytas: 0	Vienas iš: • 0 — PNG • 1 — JPG
Quality (JPG only) Pasirinktinis	`QUALITY`	[skaičius: sveikas skaičius] Default: 75	Minimum 1, maximum 100.

continues on next page

Table 24.443 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Metatile size Pasirinktinis	`METATILESIZE`	[skaičius: sveikas skaičius] Default: 4	Specify a custom metatile size when generating XYZ tiles. Larger values may speed up the rendering of tiles and provide better labelling (fewer gaps without labels) at the expense of using more memory. Minimum 1, maximum 20.
Tile width Pasirinktinis	`TILE_WIDTH`	[skaičius: sveikas skaičius] Default: 256	Minimum 1, maximum 4096.
Tile height Pasirinktinis	`TILE_HEIGHT`	[skaičius: sveikas skaičius] Default: 256	Minimum 1, maximum 4096.
Use inverted tile Y axis (TMS conventions)	`TMS_CONVENTION`	[boolean] Numatytas: ne	
Išvesties aplankas	`OUTPUT_DIRECTO`	[aplankas] Numatytas: `[Įrašyti į laikiną aplanką]`	Specification of the output directory (for the tiles). *One of*: • Praleisti išvestį • Įrašyti į laikiną aplanką • Įrašyti į aplanką
Output html (Leaflet) Pasirinktinis	`OUTPUT_HTML`	[html] Numatytas: `[Įrašyti į laikiną failą]`	Specification of the output HTML file. *One of*: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Leaflet HTML output title Pasirinktinis	`HTML_TITLE`	[tekstas] Numatytas: jokio	HTML <title>-tag used for the Leaflet HTML output file.
Leaflet HTML output attribution Pasirinktinis	`HTML_ATTRIBUTI`	[tekstas] Numatytas: jokio	Custom map attribution used for the Leaflet HTML output file. HTML links are possible.
Include OpenStreetMap basemap in Leaflet HTML output	`HTML_OSM`	[boolean] Numatytas: ne	An OpenStreetMap basemap layer (source: https://tile.openstreetmap.org) is included in the Leaflet HTML output file. Proper map attribution is added automatically.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties aplankas	`OUTPUT_DIRECTO`	[aplankas]	Output directory (for the tiles)
Output html (Leaflet)	`OUTPUT_HTML`	[html]	The output HTML (Leaflet) file

Pythono kodas

Algorithm ID: `native:tilesxyzdirectory`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Generate XYZ tiles (MBTiles)

Generates raster "XYZ" tiles using the current QGIS project as a single file in the "MBTiles" format.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Apimtis (xmin, xmax, ymin, ymax)	`EXTENT`	[apimtis]	Specify the extent of the tiles. It will internally be extended to a multiple of the tile size. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Minimum zoom	`ZOOM_MIN`	[skaičius: sveikas skaičius] Default: 12	Minimum 0, maximum 25.
Maximum zoom	`ZOOM_MAX`	[skaičius: sveikas skaičius] Default: 12	Minimum 0, maximum 25.

continues on next page

Table 24.446 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
DPI	`DPI`	[skaičius: sveikas skaičius] Default: 96	Minimum 48, maximum 600.
Background color Pasirinktinis	`BACKGROUND_COL`	[color] Default: QColor(0, 0, 0, 0)	Choose the background color for the tiles
Enable antialiasing	`ANTIALIAS`	[boolean] Numatytas: Taip	Determines if antialiasing should be enabled
Tile format	`TILE_FORMAT`	[sąrašas] Numatytas: 0	Vienas iš: • 0 — PNG • 1 — JPG
Quality (JPG only) Pasirinktinis	`QUALITY`	[skaičius: sveikas skaičius] Default: 75	Minimum 1, maximum 100.
Metatile size Pasirinktinis	`METATILESIZE`	[skaičius: sveikas skaičius] Default: 4	Specify a custom metatile size when generating XYZ tiles. Larger values may speed up the rendering of tiles and provide better labelling (fewer gaps without labels) at the expense of using more memory. Minimum 1, maximum 20.
Output file (for MBTiles)	`OUTPUT_FILE`	[file] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties failo specifikacija. *Vienas iš*: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Output file (for MBTiles)	`OUTPUT_FILE`	[file]	The output file.

Pythono kodas

Algorithm ID: `native:tilesxyzmbtiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.22 Vektorių analizė

Bazinė laukų statistika

Kuria bazinę vektorinio sluoksnio atributų lentelės lauko statistiką. Palaikomi skaitiniai, datos, laiko ir tekstiniai laukai. Grąžinama statistika priklauso nuo lauko tipo.

Statistiką galima kurti kaip lentelę arba kaip HTML failą, o pasiekti ją galima per meniu *Apdorojimas* ► *Rezultatų žiūryklė*.

Numatytas meniu: *Vektorius* ► *Analizės įrankiai*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties vektorius	`INPUT_LAYER`	[vektorius: bet koks]	Vektorinis sluoksnis, kurio statistiką reikia skaičiuoti
Laukas, kurio statistiką reikia skaičiuoti	`FIELD_NAME`	[lentelės laukas: bet koks]	Bet koks palaikomas lentelės laukas, kurio statistiką reikia skaičiuoti
Statistika Pasirinktinis	`OUTPUT`	[vektorius: lentelė] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties lentelę sukurtai statistikai. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Statistikos ataskaita Pasirinktinis	`OUTPUT_HTML_FI`	[html] Numatytas: `[Įrašyti į laikiną failą]`	Paskaičiuotos statistikos failo specifikacija. *Vienas iš*: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Statistika	`OUTPUT`	[vektorius: lentelė]	Lentelė su paskaičiuota statistika
Statistikos ataskaita	`OUTPUT_HTML_FI`	[html]	HTML failas su paskaičiuota statistika
Skaičius	`COUNT`	[skaičius: sveikas skaičius]	
Unikalių reikšmių skaičius	`UNIQUE`	[skaičius: sveikas skaičius]	
Tuščių (null) reikšmių skaičius	`EMPTY`	[skaičius: sveikas skaičius]	
Netuščių reikšmių skaičius	`FILLED`	[skaičius: sveikas skaičius]	
Minimali reikšmė	`MIN`	[toks pat, kaip įvestis]	

continues on next page

Table 24.449 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Maksimali reikšmė	`MAX`	[toks pat, kaip įvestis]	
Minimalus ilgis	`MIN_LENGTH`	[skaičius: sveikas skaičius]	
Maksimalus ilgis	`MAX_LENGTH`	[skaičius: sveikas skaičius]	
Vidutinis ilgis	`MEAN_LENGTH`	[skaičius: slankaus kablelio]	
Variacijos koeficientas	`CV`	[skaičius: slankaus kablelio]	
Suma	`SUM`	[skaičius: slankaus kablelio]	
Vidutinė reikšmė	`MEAN`	[skaičius: slankaus kablelio]	
Standartinis nuokrypis	`STD_DEV`	[skaičius: slankaus kablelio]	
Diapazonas	`RANGE`	[skaičius: slankaus kablelio]	
Mediana	`MEDIAN`	[skaičius: slankaus kablelio]	
Mažuma (rečiausiai pasitaikanti reikšmė)	`MINORITY`	[toks pat, kaip įvestis]	
Dauguma (dažniausiai pasitaikanti reikšmė)	`MAJORITY`	[toks pat, kaip įvestis]	
Pirmas kvartilis	`FIRSTQUARTILE`	[skaičius: slankaus kablelio]	
Trečias kvartilis	`THIRDQUARTILE`	[skaičius: slankaus kablelio]	
Tarpkvartilinis diapazonas (IQR)	`IQR`	[skaičius: slankaus kablelio]	

Pythono kodas

Algorithm ID: `native:basicstatisticsforfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Pakilimas linija

Skaičiuoja bendrą pakilimą ar nusileidimą palei linijos geometriją. Įvesties sluoksnis turi turėti Z reikšmes. Jei Z reikšmių nėra, galima naudoti algoritmą *Drape (set Z value from raster)* Z reikšmių užpildymui pagal DEM sluoksnį.

Išvesties sluoksnis yra įvesties sluoksnio kopija su papildomai laukais, kuriuose yra bendras pakilimas (`climb`), bendras nusileidimas (`descent`), minimalus aukštis (`minelev`) ir maksimalus aukštis (`maxelev`) kiekvienai linijos geometrijai. Jei įvesties sluoksnyje yra laukai su tokiais pačiais pavadinimais, kaip nurodyti, jie bus pervadinti (laukų pavadinimai bus pakeisti į „pavadinimas_2“, „pavadinimas_3“, ir t.t., randant pirmą nepasikartojantį pavadinimą).

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Linijų sluoksnis	`INPUT`	[vektorius: linija]	Linijų sluoksnis, kuriam reikia skaičiuoti pakilimą. Privalo turėti Z reikšmes
Pakilimo sluoksnis	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Išvesties (linijų) sluoksnio specifikacija. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Pakilimo sluoksnis	`OUTPUT`	[vektorius: linija]	Linijų sluoksnis su naujais pakilimo skaičiavimo atributais.
Bendras pakilimas	`TOTALCLIMB`	[skaičius: slankaus kablelio]	Pakilimo suma visoms įvesties sluoksnio geometrijoms
Bendras nusileidimas	`TOTALDESCENT`	[skaičius: slankaus kablelio]	Nusileidimo suma visoms įvesties sluoksnio geometrijoms
Minimalus aukštis	`MINELEVATION`	[skaičius: slankaus kablelio]	Minimalus sluoksnio geometrijų aukštis
Maksimalus aukštis	`MAXELEVATION`	[skaičius: slankaus kablelio]	Maksimalus sluoksnio geometrijų aukštis

Pythono kodas

Algorithm ID: `native:climbalongline`

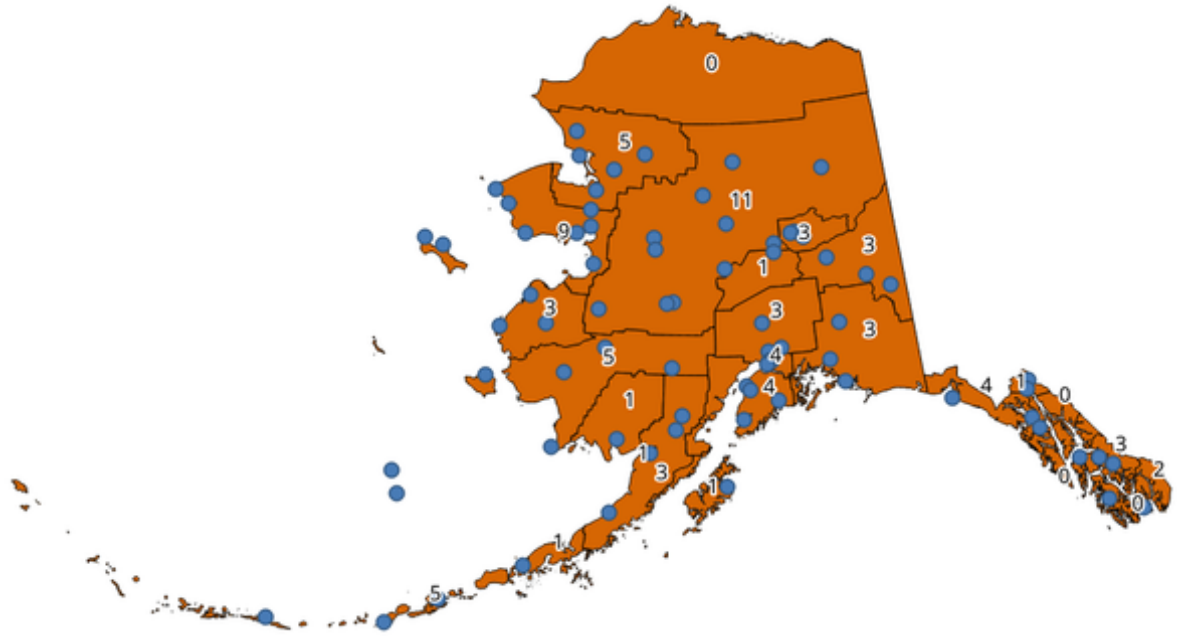
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Skaičiuoti taškus poligone

Ima taškų ir poligonų sluoksnį ir suskaičiuoja, kiek taškų sluoksnyje yra taškų, kurie yra kiekviename poligono sluoksnio poligone.

Sukuriamas naujas poligonų sluoksnis su lygiai tokiu pačiu turiniu, kaip ir įvesties poligonų sluoksnis, tik jame yra papildomas laukas su kiekvieno poligono taškų skaičiumi.



24.70 Fig. : Užrašai poligonuose rodo taškų skaičių

Galima naudoti neprivalomą svorio lauką, kuriam bus priskirtas kiekvieno taško svoris. Alternatyviai, galima nurodyti unikalios klasės lauką. Jei naudojami abu variantai, svorio laukas bus naudojamas, o unikalios klasės laukas bus ignoruojamas.

Leidžia *vietoje keisti* poligonų geoobjektus

`Numatytas meniu`: *Vektorius* ► *Analizės įrankiai*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Poligonai	`POLYGONS`	[vektorius: poligonas]	Poligonų sluoksnis, kurio geoobjektai susiejami su į juos patenkančių taškų skaičiumi
Taškai	`POINTS`	[vektorius: taškas]	Taškų sluoksnis su skaičiuojamais geoobjektais
Svorio laukas Pasirinktinis	`WEIGHT`	[lentelės laukas: bet koks]	Taškų sluoksnio laukas. Bus paskaičiuota suma svorio lauko reikšmių iš taškų, patenkančių į poligoną. Jei svoris nėra skaitinis, skaičius bus `0`.
Klasės laukas Pasirinktinis	`CLASSFIELD`	[lentelės laukas: bet koks]	Taškai klasifikuojami pagal parinktą atributą, o jei keli taškai su ta pačia atributo reikšme yra poligone, tik vienas iš jų bus paskaičiuotas. Taip paskaičiuojamas galutinis taškų su skirtingomis klasėmis poligone skaičius.
Skaičiaus lauko pavadinimas	`FIELD`	[tekstas] Numatytas: ‚NUMPOINTS‘	Pavadinimas lauko, į kurį reikia įrašyti taškų skaičių

continues on next page

Table 24.452 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Skaičius	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Išvesties sluoksnio specifikacija. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Skaičius	`OUTPUT`	[vektorius: poligonas]	Gaunamas sluoksnis su atributų lentele, kurioje yra naujas laukas su taškų skaičiumi

Pythono kodas

Algoritmo ID: `native:countpointsinpolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

DBSCAN clustering

Clusters point features based on a 2D implementation of Density-based spatial clustering of applications with noise (DBSCAN) algorithm.

The algorithm requires two parameters, a minimum cluster size, and the maximum distance allowed between clustered points.

Taip pat žiūrėkite

ST-DBSCAN clustering, *K-means clustering*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: taškas]	Layer to analyze

continues on next page

Table 24.454 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Minimum cluster size	`MIN_SIZE`	[skaičius: sveikas skaičius] Numatytas: 5	Minimum number of features to generate a cluster
Maximum distance between clustered points	`EPS`	[skaičius: slankaus kablelio] Numatytas: 1.0	Distance beyond which two features can not belong to the same cluster (eps)
Clusters	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the vector layer for the result of the clustering. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Treat border points as noise (DBSCAN*)	`DBSCAN*`	[boolean] Numatytas: ne	If checked, points on the border of a cluster are themselves treated as unclustered points, and only points in the interior of a cluster are tagged as clustered.
Cluster field name	`FIELD_NAME`	[tekstas] Default: ‚CLUSTER_ID‘	Name of the field where the associated cluster number shall be stored
Cluster size field name	`SIZE_FIELD_NAM`	[tekstas] Default: ‚CLUSTER_SIZE‘	Name of the field with the count of features in the same cluster

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Clusters	`OUTPUT`	[vektorius: taškas]	Vector layer containing the original features with a field setting the cluster they belong to
Number of clusters	`NUM_CLUSTERS`	[skaičius: sveikas skaičius]	The number of clusters discovered

Pythono kodas

Algorithm ID: `native:dbscanclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Distance matrix

Calculates for point features distances to their nearest features in the same layer or in another layer.

Numatytas meniu: *Vektorius ► Analizės įrankiai*

Taip pat žiūrėkite

Join attributes by nearest

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input point layer	`INPUT`	[vektorius: taškas]	Point layer for which the distance matrix is calculated (**from** points)
Input unique ID field	`INPUT_FIELD`	[lentelės laukas: bet koks]	Field to use to uniquely identify features of the input layer. Used in the output attribute table.
Target point layer	`TARGET`	[vektorius: taškas]	Point layer containing the nearest point(s) to search (**to** points)
Target unique ID field	`TARGET_FIELD`	[lentelės laukas: bet koks]	Field to use to uniquely identify features of the target layer. Used in the output attribute table.
Output matrix type	`MATRIX_TYPE`	[sąrašas] Numatytas: 0	Different types of calculation are available: • 0 — Linear (N * *k* x 3) distance matrix: for each input point, reports the distance to each of the *k* nearest target points. The output matrix consists of up to *k* rows per input point, and each row has three columns: *InputID*, *TargetID* and *Distance*. • 1 — Standard (N x T) distance matrix • 2 — Summary distance matrix (mean, std. dev., min, max): for each input point, reports statistics on the distances to its target points.
Use only the nearest (k) target points	`NEAREST_POINTS`	[skaičius: sveikas skaičius] Numatytas: 0	You can choose to calculate the distance to all the points in the target layer (*0*) or limit to a number (*k*) of closest features.
Distance matrix	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Išvesties vektorinio sluoksnio specifikacija. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Distance matrix	`OUTPUT`	[vektorius: taškas]	Point (or MultiPoint for the „Linear (N * *k* x 3)“ case) vector layer containing the distance calculation for each input feature. Its features and attribute table depend on the selected output matrix type.

Pythono kodas

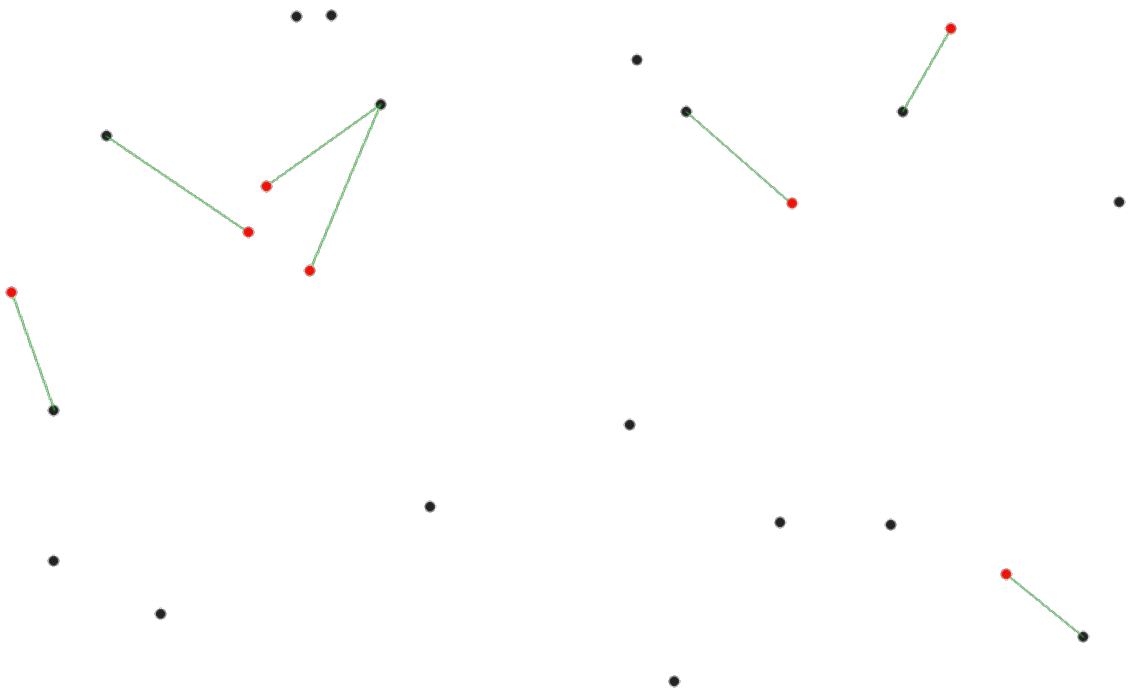
Algorithm ID: `qgis:distancematrix`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Distance to nearest hub (line to hub)

Creates lines that join each feature of an input vector to the nearest feature in a destination layer. Distances are calculated based on the *center* of each feature.



24.71 Fig. : Display the nearest hub for the red input features

Taip pat žiūrėkite

Distance to nearest hub (points), *Join attributes by nearest*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Source points layer	`INPUT`	[vektorius: geometrija]	Vector layer for which the nearest feature is searched
Destination hubs layer	`HUBS`	[vektorius: geometrija]	Vector layer containing the features to search for
Hub layer name attribute	`FIELD`	[lentelės laukas: bet koks]	Field to use to uniquely identify features of the destination layer. Used in the output attribute table
Measurement unit	`UNIT`	[sąrašas] Numatytas: 0	Units in which to report the distance to the closest feature: • 0 — Meters • 1 — Feet • 2 — Miles • 3 — Kilometers • 4 — Layer units
Hub distance	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output line vector layer connecting the matching points. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Hub distance	`OUTPUT`	[vektorius: linija]	Line vector layer with the attributes of the input features, the identifier of their closest feature and the calculated distance.

Pythono kodas

Algorithm ID: `qgis:distancetonearesthublinetohub`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Distance to nearest hub (points)

Creates a point layer representing the *center* of the input features with the addition of two fields containing the identifier of the nearest feature (based on its center point) and the distance between the points.

Taip pat žiūrėkite

Distance to nearest hub (line to hub), *Join attributes by nearest*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Source points layer	`INPUT`	[vektorius: bet koks]	Vector layer for which the nearest feature is searched
Destination hubs layer	`HUBS`	[vektorius: bet koks]	Vector layer containing the features to search for
Hub layer name attribute	`FIELD`	[lentelės laukas: bet koks]	Field to use to uniquely identify features of the destination layer. Used in the output attribute table
Measurement unit	`UNIT`	[sąrašas] Numatytas: 0	Units in which to report the distance to the closest feature: • 0 — Meters • 1 — Feet • 2 — Miles • 3 — Kilometers • 4 — Layer units
Hub distance	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output point vector layer with the nearest hub. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Hub distance	`OUTPUT`	[vektorius: taškas]	Point vector layer representing the center of the source features with their attributes, the identifier of their closest feature and the calculated distance.

Pythono kodas

Algorithm ID: `qgis:distancetonearesthubpoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Join by lines (hub lines)

Creates hub and spoke diagrams by connecting lines from points on the Spoke layer to matching points in the Hub layer.

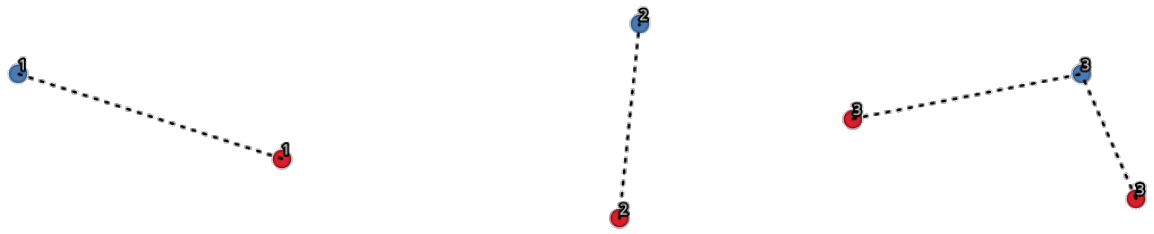
Determination of which hub goes with each point is based on a match between the Hub ID field on the hub points and the Spoke ID field on the spoke points.

If input layers are not point layers, a point on the surface of the geometries will be taken as the connecting location.

Optionally, geodesic lines can be created, which represent the shortest path on the surface of an ellipsoid. When geodesic mode is used, it is possible to split the created lines at the antimeridian (±180 degrees longitude), which can improve rendering of the lines. Additionally, the distance between vertices can be specified. A smaller distance results in a denser, more accurate line.

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.



24.72 Fig. : Join points based on a common field / attribute

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Hub layer	`HUBS`	[vektorius: geometrija]	Input layer
Hub ID field	`HUB_FIELD`	[lentelės laukas: bet koks]	Field of the hub layer with ID to join

continues on next page

Table 24.463 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Hub layer fields to copy (leave empty to copy all fields) Pasirinktinis	`HUB_FIELDS`	[lentelės laukas: bet koks] [sąrašas]	The field(s) of the hub layer to be copied. If no field(s) are chosen all fields are taken.
Spoke layer	`SPOKES`	[vektorius: geometrija]	Additional spoke point layer
Spoke ID field	`SPOKE_FIELD`	[lentelės laukas: bet koks]	Field of the spoke layer with ID to join
Spoke layer fields to copy (leave empty to copy all fields) Pasirinktinis	`SPOKE_FIELDS`	[lentelės laukas: bet koks] [sąrašas]	Field(s) of the spoke layer to be copied. If no fields are chosen all fields are taken.
Create geodesic lines	`GEODESIC`	[boolean] Numatytas: ne	Create geodesic lines (the shortest path on the surface of an ellipsoid)
Hub lines	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output hub line vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Distance between vertices (geodesic lines only)	`GEODESIC_DISTA`	[skaičius: slankaus kablelio] Default: 1000.0 (kilometers)	Distance between consecutive vertices (in kilometers). A smaller distance results in a denser, more accurate line
Split lines at antimeridian (±180 degrees longitude)	`ANTIMERIDIAN_S`	[boolean] Numatytas: ne	Split lines at ±180 degrees longitude (to improve rendering of the lines)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Hub lines	`OUTPUT`	[vektorius: linija]	The resulting line layer connecting matching points in input layers

Pythono kodas

Algorithm ID: `native:hublines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

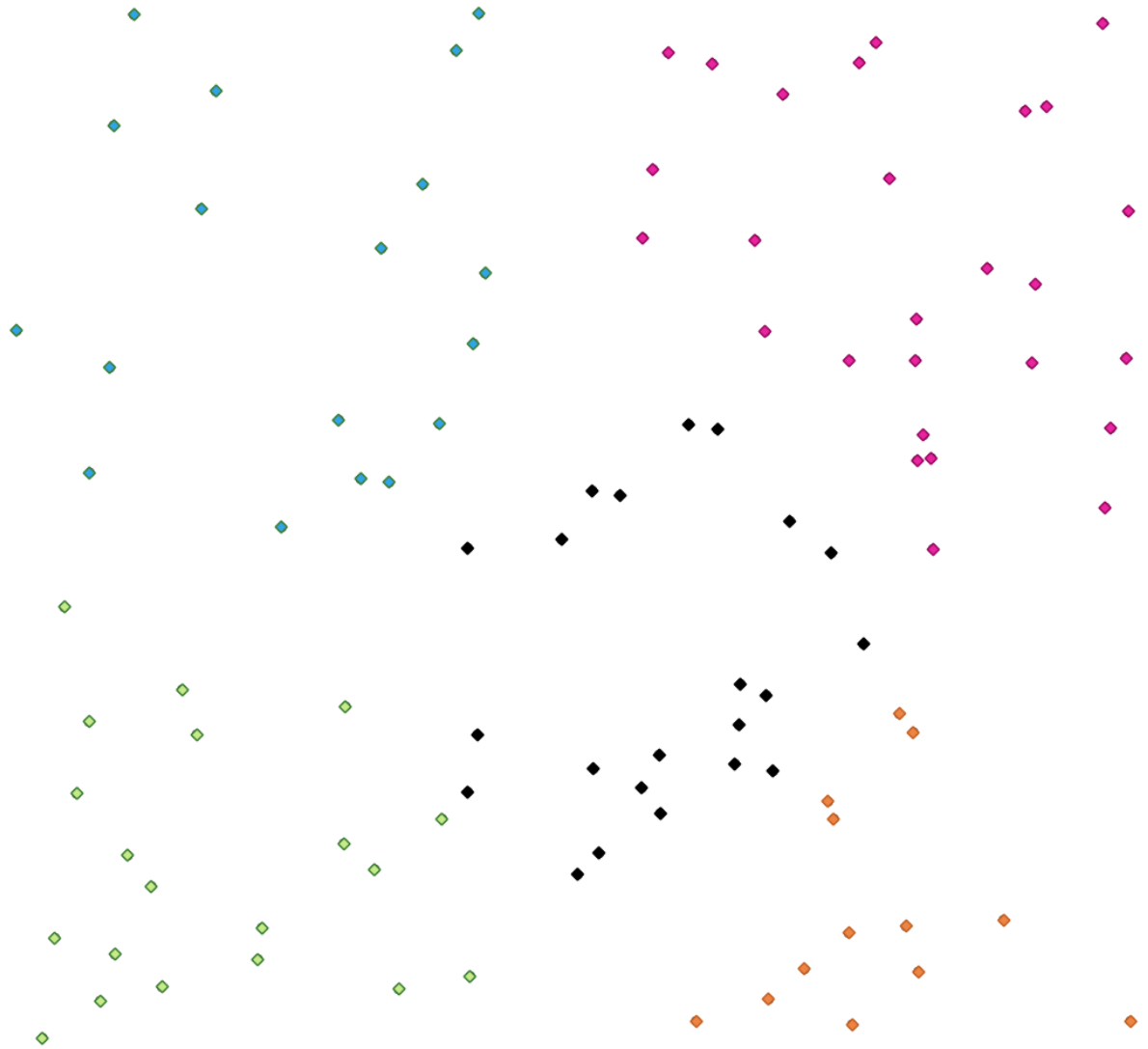
algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

K-means clustering

Calculates the 2D distance based k-means cluster number for each input feature.

K-means clustering aims to partition the features into k clusters in which each feature belongs to the cluster with the nearest mean. The mean point is represented by the barycenter of the clustered features.

If input geometries are lines or polygons, the clustering is based on the centroid of the feature.



24.73 Fig. : A five class point clusters

Taip pat žiūrėkite

DBSCAN clustering, *ST-DBSCAN clustering*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Layer to analyze
Number of clusters	`CLUSTERS`	[skaičius: sveikas skaičius] Numatytas: 5	Number of clusters to create with the features
Metodas	`METHOD`	[sąrašas] Numatytas: 0	Initial centers selection method. Possible values are: • 0 — Farthest points: the initial centers are selected as the farthest points from each other. • 1 — K-means++: The first center is selected uniformly at random from the set of input features. Subsequent centers are chosen based on a weighted probability distribution that favors features farther from existing centers. This ensures that the initial centers are well-dispersed, reducing the likelihood of poor clustering and improving the algorithm's performance. *References: Arthur, David & Vassilvitskii, Sergei. (2007). K-Means++: The Advantages of Careful Seeding. Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms. 8.*
Clusters	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specify the output vector layer for the generated clusters. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Cluster field name	`FIELD_NAME`	[tekstas] Default: ‚CLUSTER_ID‘	Name of the field where the associated cluster number shall be stored

continues on next page

Table 24.467 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Cluster size field name	`SIZE_FIELD_NAM`	[tekstas] Default: ‚CLUSTER_SIZE‘	Name of the field with the count of features in the same cluster

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Clusters	`OUTPUT`	[vector: same as input]	Vector layer containing the original features with fields specifying the cluster they belong to and their number in it

Pythono kodas

Algorithm ID: `native:kmeansclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

List unique values

Lists unique values of an attribute table field and counts their number.

Numatytas meniu: *Vektorius* ► *Analizės įrankiai*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Layer to analyze
Target field(s)	`FIELDS`	[lentelės laukas: bet koks] [sąrašas]	Field(s) to analyze
Unique values Pasirinktinis	`OUTPUT`	[vektorius: lentelė] Default: `[Create temporary layer]`	Specify the summary table layer with unique values. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

continues on next page

Table 24.469 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
HTML report Pasirinktinis	`OUTPUT_HTML_FI`	[html] Default: `[Save to temporary file]`	HTML report of unique values in the *Processing* ► *Results viewer*. *One of*: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Unique values	`OUTPUT`	[vektorius: lentelė]	Summary table layer with unique values
HTML report	`OUTPUT_HTML_FI`	[html]	HTML report of unique values. Can be opened from the *Processing* ► *Results viewer*
Total unique values	`TOTAL_VALUES`	[skaičius: sveikas skaičius]	The number of unique values in the input field
Unique values concatenated	`UNIQUE_VALUES`	[tekstas]	A string with the comma separated list of unique values found in the input field

Pythono kodas

Algorithm ID: `native:listuniquevalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Mean coordinate(s)

Computes a point layer with the center of mass of geometries in an input layer.

An attribute can be specified as containing weights to be applied to each feature when computing the center of mass.

If an attribute is selected in the parameter, features will be grouped according to values in this field. Instead of a single point with the center of mass of the whole layer, the output layer will contain a center of mass for the features in each category.

Numatytas meniu: *Vektorius* ► *Analizės įrankiai*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Svorio laukas Pasirinktinis	`WEIGHT`	[lentelės laukas: skaičius]	Field to use if you want to perform a weighted mean
Unique ID field	`UID`	[lentelės laukas: bet koks]	Unique field on which the calculation of the mean will be made
Mean coordinates	`OUTPUT`	[vektorius: taškas] Default: `[Create temporary layer]`	Specify the (point vector) layer for the result. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Mean coordinates	`OUTPUT`	[vektorius: taškas]	Resulting point(s) layer

Pythono kodas

Algorithm ID: `native:meancoordinates`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Nearest neighbour analysis

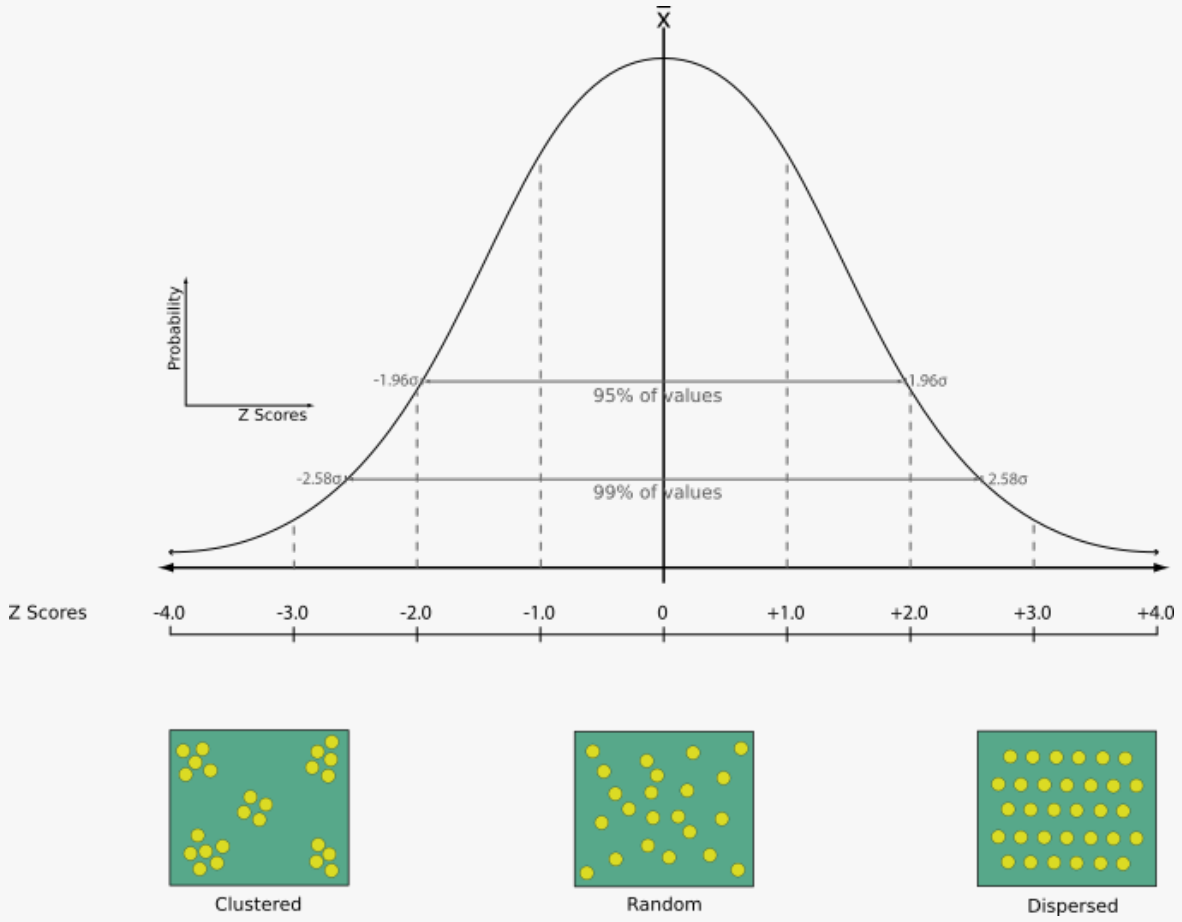
Performs nearest neighbor analysis for a point layer. The output tells you how your data are distributed (clustered, randomly or distributed).

Output is generated as an HTML file with the computed statistical values:

- Observed mean distance
- Expected mean distance
- Nearest neighbour index
- Number of points
- Z-Score: Comparing the Z-Score with the normal distribution tells you how your data are distributed. A low Z-Score means that the data are unlikely to be the result of a spatially random process, while a high Z-Score means that your data are likely to be a result of a spatially random process.

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.



Numatytas meniu: *Vektorius* ► *Analizės įrankiai*

Taip pat žiūrėkite

Join attributes by nearest

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: taškas]	Point vector layer to calculate the statistics on
Nearest neighbour Pasirinktinis	`OUTPUT_HTML_FI`	[html] Default: `[Save to temporary file]`	Specification of the HTML file for the computed statistics. *One of*: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Nearest neighbour	`OUTPUT_HTML_FI`	[html]	HTML file with the computed statistics
Observed mean distance	`OBSERVED_MD`	[skaičius: slankaus kablelio]	Observed mean distance
Expected mean distance	`EXPECTED_MD`	[skaičius: slankaus kablelio]	Expected mean distance
Nearest neighbour index	`NN_INDEX`	[skaičius: sveikas skaičius]	Nearest neighbour index
Number of points	`POINT_COUNT`	[skaičius: sveikas skaičius]	Number of points
Z-Score	`Z_SCORE`	[skaičius: slankaus kablelio]	Z-Score

Pythono kodas

Algorithm ID: `native:nearestneighbouranalysis`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Overlap analysis

Calculates the area and percentage cover by which features from an input layer are overlapped by features from a selection of overlay layers.

New attributes are added to the output layer reporting the total area of overlap and percentage of the input feature overlapped by each of the selected overlay layers.

> **Pastaba**
>
> Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Įvesties sluoksnis.
Perdengimo sluoksniai	`LAYERS`	[vector: polygon] [list]	The overlay layers.

continues on next page

Table 24.475 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Overlap	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Tinklelio dydis Pasirinktinis	`GRID_SIZE`	[skaičius: slankaus kablelio] Numatytas: jokio	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Overlap	`OUTPUT`	[toks pat, kaip įvestis]	The output layer with additional fields reporting the overlap (in map units and percentage) of the input feature overlapped by each of the selected layers.

Pythono kodas

Algorithm ID: `native:calculatevectoroverlaps`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

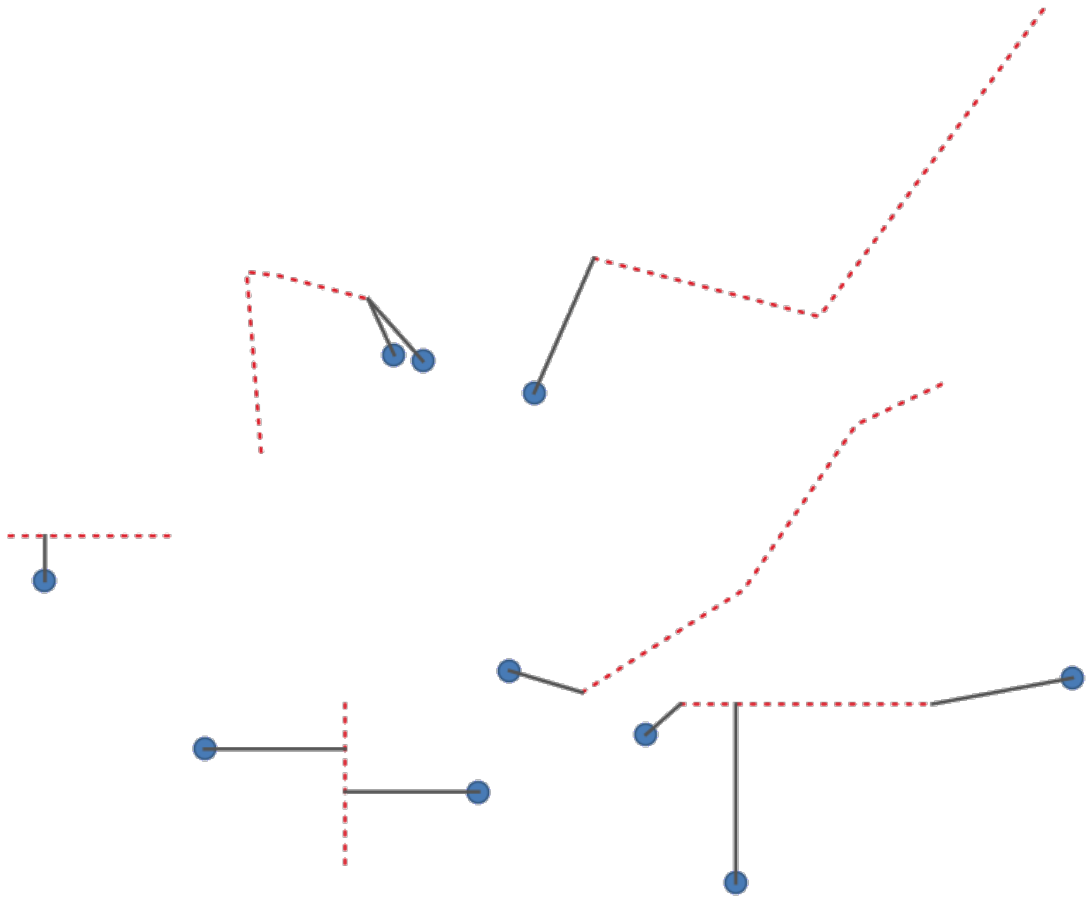
Shortest line between features

Creates a line layer as the shortest line between the source and the destination layer. By default only the first nearest feature of the destination layer is taken into account. The n-nearest neighboring features number can be specified. If a maximum distance is specified, then only features which are closer than this distance will be considered.

The output features will contain all the source layer attributes, all the attributes from the n-nearest feature and the additional field of the distance.

Svarbu

This algorithm uses purely Cartesian calculations for distance, and does not consider geodetic or ellipsoid properties when determining feature proximity. The measurement and output coordinate system is based on the coordinate system of the source layer.



24.74 Fig. : Shortest line from point features to lines

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Source layer	`SOURCE`	[vektorius: geometrija]	Origin layer for which to search for nearest neighbors
Destination layer	`DESTINATION`	[vektorius: geometrija]	Target Layer in which to search for nearest neighbors
Metodas	`METHOD`	[sąrašas] Numatytas: 0	Shortest distance calculation method Possible values are: • 0 — Distance to nearest point on feature • 1 — Distance to feature centroid
Maximum number of neighbors	`NEIGHBORS`	[skaičius: sveikas skaičius] Numatytas: 1	Maximum number of neighbors to look for

continues on next page

Table 24.478 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Maximum distance Pasirinktinis	`DISTANCE`	[skaičius: slankaus kablelio]	Only destination features which are closer than this distance will be considered.
Shortest lines	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	`OUTPUT`	[vektorius: linija]	Line vector layer joining source features to their nearest neighbor(s) in the destination layer. Contains all attributes for both source and destination features, and the computed distance.

Pythono kodas

Algorithm ID: `native:shortestline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

ST-DBSCAN clustering

Clusters point features based on a 2D implementation of spatiotemporal Density-based clustering of applications with noise (ST-DBSCAN) algorithm.

Taip pat žiūrėkite

DBSCAN clustering, *K-means clustering*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: taškas]	Layer to analyze
Date/time field	`DATETIME_FIELD`	[tablefield: date]	Field containing the temporal information
Minimum cluster size	`MIN_SIZE`	[skaičius: sveikas skaičius] Numatytas: 5	Minimum number of features to generate a cluster
Maximum distance between clustered points	`EPS`	[skaičius: slankaus kablelio] Numatytas: 1.0	Distance beyond which two features can not belong to the same cluster (eps)
Maximum time duration between clustered points	`EPS2`	[skaičius: slankaus kablelio] Default: 0.0 (days)	Time duration beyond which two features can not belong to the same cluster (eps2). Available time units are milliseconds, seconds, minutes, hours, days and weeks.
Clusters	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the vector layer for the result of the clustering. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Treat border points as noise (DBSCAN*)	`DBSCAN*`	[boolean] Numatytas: ne	If checked, points on the border of a cluster are themselves treated as unclustered points, and only points in the interior of a cluster are tagged as clustered.
Cluster field name	`FIELD_NAME`	[tekstas] Default: ‚CLUSTER_ID‘	Name of the field where the associated cluster number shall be stored
Cluster size field name	`SIZE_FIELD_NAM`	[tekstas] Default: ‚CLUSTER_SIZE‘	Name of the field with the count of features in the same cluster

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Clusters	`OUTPUT`	[vektorius: taškas]	Vector layer containing the original features with a field setting the cluster they belong to
Number of clusters	`NUM_CLUSTERS`	[skaičius: sveikas skaičius]	The number of clusters discovered

Pythono kodas

Algorithm ID: `native:stdbscanclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Statistics by categories

Calculates statistics of a field depending on a parent class. The parent class is a combination of values from other fields.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input vector layer	`INPUT`	[vektorius: bet koks]	Input vector layer with unique classes and values
Field to calculate statistics on (if empty, only count is calculated) Pasirinktinis	`VALUES_FIELD_N`	[lentelės laukas: bet koks]	If empty only the count will be calculated
Field(s) with categories	`CATEGORIES_FIE`	[lentelės laukas: bet koks] [sąrašas]	The fields that (combined) define the categories
Statistics by category	`OUTPUT`	[vektorius: lentelė] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties lentelę sukurtai statistikai. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Statistics by category	`OUTPUT`	[vektorius: lentelė]	Table containing the statistics

Depending on the type of the field being analyzed, the following statistics are returned for each grouped value:

Statistika	Tekstas	Numeric	Data
Count (`COUNT`)	☑	☑	☑
Unique values (`UNIQUE`)	☑		☑

continues on next page

Table 24.485 – tęsinys iš praeito puslapio

Statistika	Tekstas	Numeric	Data
Empty (null) values (EMPTY)	☑		☑
Non-empty values (FILLED)	☑		☑
Minimal value (MIN)	☑	☑	☑
Maximal value (MAX)	☑	☑	☑
Range (RANGE)		☑	
Sum (SUM)		☑	
Mean value (MEAN)		☑	
Median value (MEDIAN)		☑	
Standard Deviation (STD_DEV)		☑	
Coefficient of variation (CV)		☑	
Minority (rarest occurring value - MINORITY)		☑	
Majority (most frequently occurring value - MAJORITY)		☑	
First Quartile (FIRSTQUARTILE)		☑	
Third Quartile (THIRDQUARTILE)		☑	
Inter Quartile Range (IQR)		☑	
Minimum Length (MIN_LENGTH)	☑		
Mean Length (MEAN_LENGTH)	☑		
Maximum Length (MAX_LENGTH)	☑		

Pythono kodas

Algorithm ID: `qgis:statisticsbycategories`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Sum line lengths

Takes a polygon layer and a line layer and measures the total length of lines and the total number of them that cross each polygon.

The resulting layer has the same features as the input polygon layer, but with two additional attributes containing the length and count of the lines across each polygon.

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.

☑ Leidžia *vietoje keisti* poligonų geoobjektus

Numatytas meniu: *Vektorius* ► *Analizės įrankiai*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Lines	`LINES`	[vektorius: linija]	Input vector line layer
Poligonai	`POLYGONS`	[vektorius: poligonas]	Polygon vector layer
Lines length field name	`LEN_FIELD`	[tekstas] Default: ‚LENGTH‘	Name of the field for the lines length
Lines count field name	`COUNT_FIELD`	[tekstas] Default: ‚COUNT‘	Name of the field for the lines count
Line length	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output polygon layer with generated statistics. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Line length	`OUTPUT`	[vektorius: poligonas]	Polygon output layer with fields of lines length and line count

Pythono kodas

Algorithm ID: `native:sumlinelengths`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.23 Vektorių padengimas

Sulieti padengimą

Veikia su padengimu (išreikštu poligonų geoobjektų aibe su tiksliai atitinkančiomis kraštų geometrijomis), kuriame suliejamos (sąjunga) geometrijos. Teikia stipriai optimizuotą būdą skaičiuoti geoobjektų sąjungą lyginant su standartiniais suliejimo įrankiais.

Taip pat žiūrėkite

Dissolve

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Įvesties poligonų vektorinis sluoksnis
Sulietas	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sulietas	`OUTPUT`	[vektorius: poligonas]	Išvesties poligonų vektorinis sluoksnis su sulietomis geometrijomis.

Pythono kodas

Algoritmo ID: `native:coverageunion`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Paprastinti padengimą

Dirba su padengimu (išreikštu aibe poligonų geoobjektų, kurių kratinių geometrijos tiksliai atitinka) pritaikant kraštinėms Visvalingam–Whyatt paprastinimą, mažinant sudėtingumą priklausomai nuo pateiktos tolerancijos, tuo pačiu išlaikant tinkamą padengimą (t.y. po paprastinimo nei viena kraštinė nesikirs ir nesilies). Geometrijos niekada nebus pašalintos, bet jos gali būti supaprastintos iki trikampio. Taipogi, kai kurios geometrijos (tokios kaip poligonai, kurie turi per mažai nepasikartojančių taškų) bus grąžinamos nepakeistos.

Jei įvesties duomenų aibė nėra tinkamas padengimas dėl persidengimų, ji vis tiek bus supaprastinta, bet netinkama topologija, tokia kaip susikertančios kraštinės, liks netinkama.

Dėmesio

Running this algorithm requires QGIS installed with GEOS >= 3.12 (see *Help* ► *About* menu).

Taip pat žiūrėkite

Simplify, *Smooth*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Įvesties poligonų vektorinis sluoksnis
Tolerancija	`TOLERANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Maksimalus atstumas (pasirinktais vienetais) tarp dviejų greta esančių viršūnių, kad būtų sulietos.
Išlaikyti ribą	`PRESERVE_BOUND`	[boolean] Numatytas: ne	Įjungus, išorinės padengimo ribos bus paliekamos nepaprastintos
Supaprastinta	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Supaprastinta	`OUTPUT`	[vektorius: poligonas]	Išvesties poligonų sluoksnis su mažesniu viršūnių skaičiumi.

Pythono kodas

Algoritmo ID: `native:coveragesimplify`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Validate coverage

Analyzes a coverage (represented as a set of polygon features with exactly matching edge geometry) to find places where the assumption of exactly matching edges is not met. Invalidity includes polygons that overlap or that have gaps smaller than the specified gap width.

Dėmesio

Running this algorithm requires QGIS installed with GEOS >= 3.12 (see *Help* ► *About* menu).

Taip pat žiūrėkite

Tikrinti tvarkingumą, *Small gaps*, *Overlaps*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Įvesties poligonų vektorinis sluoksnis
Gap width	`GAP_WIDTH`	[skaičius: slankaus kablelio] Numatytas: 0.0	The maximum width of gap to detect
Invalid edges	`INVALID_EDGES`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Invalid edges	`OUTPUT`	[vektorius: linija]	The output line vector layer showing error edges for each invalid polygon.
Validity result	`IS_VALID`	[boolean]	Returns whether the coverage is valid or not.

Pythono kodas

Algorithm ID: `native:coveragevalidate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

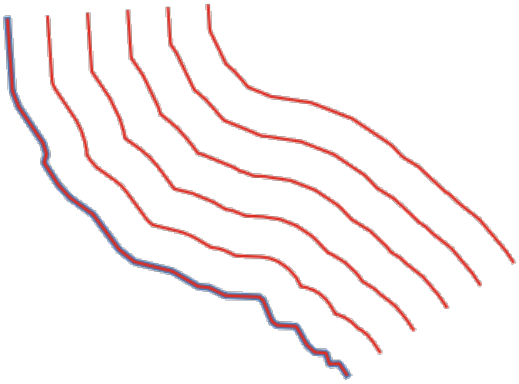
algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.24 Vektorių kūrimas

Paslinktų (lygiagrečių) linijų masyvas

Sukuria sluoksnyje linijinių geoobjektų kopijas, kuriant kelias paslinktas kiekvieno geoobjekto versijas. Kiekviena nauja versija papildomai paslenkama nurodytą atstumą.

Teigiamas atstumas paslinks linijas į kairę, o neigiami atstumai - į dešinę.



24.75 Fig. : Mėlynas - pradinis sluoksnis, raudonas - paslinktasis

Leidžia linijų *geoobjektus keisti vietoje*

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

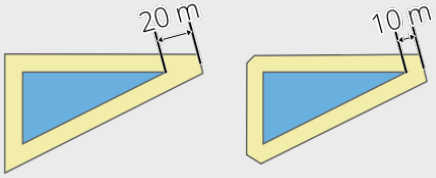
Offset lines, *Pakeistų geoobjektų masyvas*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties linijų vektorinis sluoksnis, kurį reikia naudoti poslinkiams.
Kuriamų geoobjektų skaičius	`COUNT`	[skaičius: sveikas skaičius] Numatytas: 10	Kiekvienam geoobjektui kuriamų paslinktų kopijų skaičius
Poslinkio žingsnio atstumas	`OFFSET`	[skaičius: slankaus kablelio] Numatytas: 1.0	Atstumas tarp dviejų viena po kitos einančių paslinktų kopijų
Paslinktos linijos	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties linijų sluoksnį su paslinktais geoobjektais. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Segmentai	`SEGMENTS`	[skaičius: slankaus kablelio] Numatyta: 8	Linijos segmentų skaičius, kurį naudoti aproksimuojant apskritimo ketvirtį kuriant apvalius poslinkius
Jungimo stilius	`JOIN_STYLE`	[sąrašas] Numatytas: 0	Nurodykite, kokie turi būti naudojami jungimai paslenkant linijų kampus: apvalūs, nukirsti ar nuožulnūs. Vienas iš: • 0 — Apvalūs • 1 — Mitra • 2 — Kampas 24.76 Fig. : Apvalūs, mitros ir kampo jungimo stiliai
Mitros riba	`MITER_LIMIT`	[skaičius: slankaus kablelio] Numatyta: 2.0	Nustato maksimalų atstumą nuo paslenkamos geometrijos kuriant mitros jungtį. Nurodomas kaip poslinkio atstumo koeficientas (taikoma tik mitros jungimo stiliui). Minimalus: 1.0  24.77 Fig. : 10m buferis su 2 riba ir 10m buferis su 1 riba

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Paslinktos linijos	`OUTPUT`	[vektorius: linija]	Išvesties linijų sluoksnis su pradiniais ir paslinktais geoobjektais. Atributai bus kopijuojami į atitinkamas išvestis. Bus pridėti šie, su įvesties parametrais susiję atributai: `instance` indeksuojantis `COUNT` paslinktas linijas ir `offset`, nurodantis atstumą nuo pradinės linijos.

Pythono kodas

Algoritmo ID: `native:arrayoffsetlines`

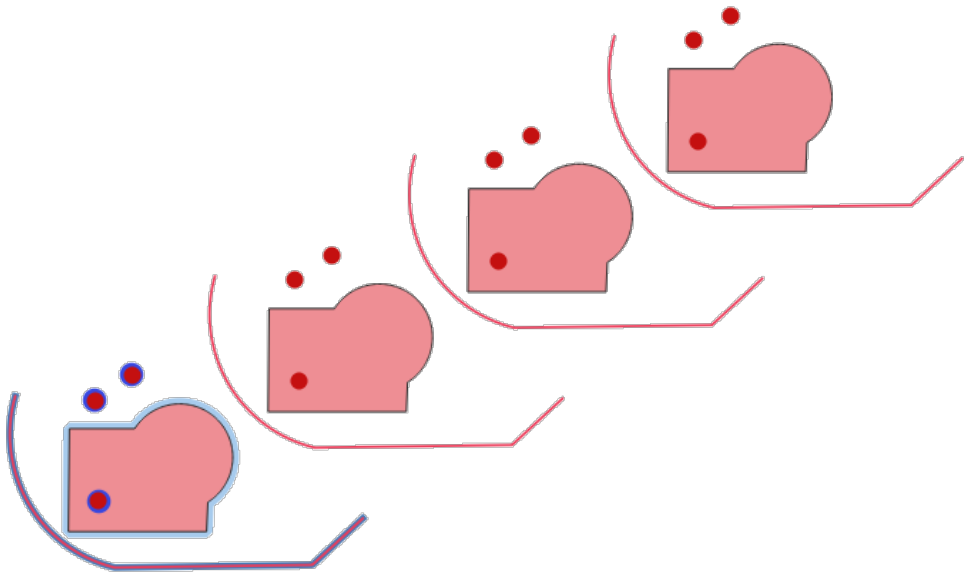
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Pakeistų geoobjektų masyvas

Sukuria sluoksnyje kelias pakeistas geoobjektų kopijas. Kiekviena kopija inkrementiškai pastumta nurodytu atstumu X, Y ir/ar Z ašyse.

Geometrijoje esančias M reikšmes taipogi galima keisti.



24.78 Fig. : Įvesties sluoksnis mėlynas, išvesties sluoksnis su pakeistais geoobjektais - raudonas

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Translate, *Paslinktų (lygiagrečių) linijų masyvas*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Keičiamas įvesties sluoksnis
Kuriamų geoobjektų skaičius	`COUNT`	[skaičius: sveikas skaičius] Numatytas: 10	Kiekvieno geoobjekto kuriamų kopijų skaičius
Žingsnio atstumas (x-ašis)	`DELTA_X`	[skaičius: slankaus kablelio] Numatytas: 0.0	X ašyje taikomas poslinkis
Žingsnio atstumas (y-ašis)	`DELTA_Y`	[skaičius: slankaus kablelio] Numatytas: 0.0	Y ašyje taikomas poslinkis
Žingsnio atstumas (z-ašis)	`DELTA_Z`	[skaičius: slankaus kablelio] Numatytas: 0.0	Z ašyje taikomas poslinkis
Žingsnio atstumas (m reikšmės)	`DELTA_M`	[skaičius: slankaus kablelio] Numatytas: 0.0	M taikomas poslinkis
Pakeista	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Išvesties linijų sluoksnis su pakeistomis (pastumtomis) geoobjektų kopijomis. Originalūs geoobjektai irgi nukopijuojami. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Pakeista	`OUTPUT`	[toks pat, kaip įvestis]	Išvesties vektorinis sluoksnis su pakeistomis (paslinktomis) geoobjektų kopijomis. Pradiniai geoobjektai irgi nukopijuojami.

Pythono kodas

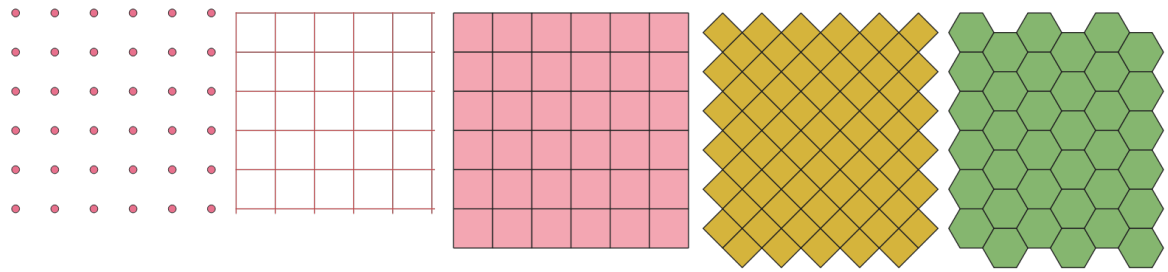
Algoritmo ID: `native:arraytranslatedfeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Kurti tinklelį

Sukuria vektorinį sluoksnį su tinkleliu, dengiančiu nurodytą apimtį. Tinklelio celės gali būti įvairių formų:



24.79 Fig. : Skirtingos tinklelio celių formos, taikomos tai pačiai apimčiai, be persidengimų

Kiekvieno tinklelio elemento dydis ir/ar vieta nurodoma naudojant horizontalų ir vertikalų atstumą. Reikia nurodyti išvesties sluoksnio CRS. Tinklelio apimties ir tarpų reikšmės turi būti išreikštos CRS koordinatėmis ir vienetais.

Numatytas meniu: *Vektorius* ► *Tyrimo įrankiai*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Tinklelio tipas	`TYPE`	[sąrašas] Numatytas: 0	Tinklelio forma. Vienas iš: • 0 — Taškas • 1 — Linija • 2 — Stačiakampis (poligonas) • 3 — Deimantas (poligonas) • 4 — Šešiakampis (poligonas)

continues on next page

Table 24.499 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Tinklelio apimtis	`EXTENT`	[apimtis]	Tinklelio apimtis Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Horizontalus dydis	`HSPACING`	[skaičius: slankaus kablelio] Numatytas: 1.0	Tinklelio celės dydis X-ašyje
Vertikalus dydis	`VSPACING`	[skaičius: slankaus kablelio] Numatytas: 1.0	Tinklelio celės dydis Y-ašyje
Horizontalus persidengimas	`HOVERLAY`	[skaičius: slankaus kablelio] Numatytas: 0.0	Persidengimo tarp dviejų greta esančių tinklelio celių dydis X ašyje.
Vertikalus persidengimas	`VOVERLAY`	[skaičius: slankaus kablelio] Numatytas: 0.0	Persidengimo tarp dviejų greta esančių tinklelio celių dydis Y ašyje.
Tinklelio CRS	`CRS`	[crs] Numatyta: *Projekto CRS*	Tinkleliui taikoma koordinačių atskaitos sistema
Tinklelis	`OUTPUT`	[vektorius: geometrija] Numatytas: `[Kurti laikiną sluoksnį]`	Gautas vektorinis tinklelio sluoksnis. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Tinklelis	`OUTPUT`	[vektorius: geometrija]	Gautas vektorinis tinklelio sluoksnis. Išvesties geometrijų tipas (taškas, linija ar poligonas) priklauso nuo *Tinklelio tipo*. Geoobjektai kuriami nuo viršaus žemyn ir iš kairės į dešinę. Atributų lentelė užpildoma: • `id` • kraštų koordinatės `left`, `right`, `top` ir `bottom` • vieta tinklelyje: `row_index` ir `column_index` (pateikiama taško, stačiakampio ir šešiakampio tinklelio tipams)

Pythono kodas

Algoritmo ID: `native:creategrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Create layer from extent

Creates a new vector layer that contains a single feature with geometry matching the extent of the input layer.

It can be used in models to convert a literal extent (`xmin`, `xmax`, `ymin`, `ymax` format) into a layer which can be used for other algorithms which require a layer based input.

Taip pat žiūrėkite

Create layer from point

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Extent (xmin, xmax, ymin, ymax)	`INPUT`	[apimtis]	Input extent Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Extent	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Extent	`OUTPUT`	[vektorius: poligonas]	The output (extent) vector layer

Pythono kodas

Algorithm ID: `native:extenttolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Create layer from point

Creates a new vector layer that contains a single feature with geometry matching a point parameter. It can be used in models to convert a point into a point layer for algorithms which require a layer based input.

Taip pat žiūrėkite

Create layer from extent

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Point	`INPUT`	[coordinates]	Input point, including CRS info (example: `397254,6214446 [EPSG:32632]`). If the CRS is not provided, the Project CRS will be used. The point can be specified by clicking on the map canvas.
Point	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Point	`OUTPUT`	[vektorius: taškas]	The output point vector layer containing the input point.

Pythono kodas

Algorithm ID: `native:pointtolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Kurti taškų sluoksnį iš lentelės

Kuria taškų sluoksnį iš lentelės su stulpeliais, turinčiais koordinačių laukus.

Be X ir Y koordinačių jūs dar galite nurodyti Z ir M laukus.

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vector: any]	Įvesties vektorinis sluoksnis arba lentelė.
X laukas	`XFIELD`	[tablefield: any]	Laukas su X koordinate.
Y laukas	`YFIELD`	[tablefield: any]	Laukas su Y koordinate.
Z laukas Pasirinktinis	`ZFIELD`	[tablefield: any]	Laukas su Z koordinate.
M laukas Pasirinktinis	`MFIELD`	[tablefield: any]	Laukas su M koordinate.
Paskirties CRS	`TARGET_CRS`	[crs] Numatytoji: `EPSG:4326`	Sluoksniui naudotina koordinačių atskaitos sistema. Daroma prielaida, kad pateiktos koordinatės suderinamos.
Taškai iš lentelės	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite gaunamą taškų sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Taškai iš lentelės	`OUTPUT`	[vektorius: taškas]	Gaunamas taškų sluoksnis

Pythono kodas

Algoritmo ID: `native:createpointslayerfromtable`

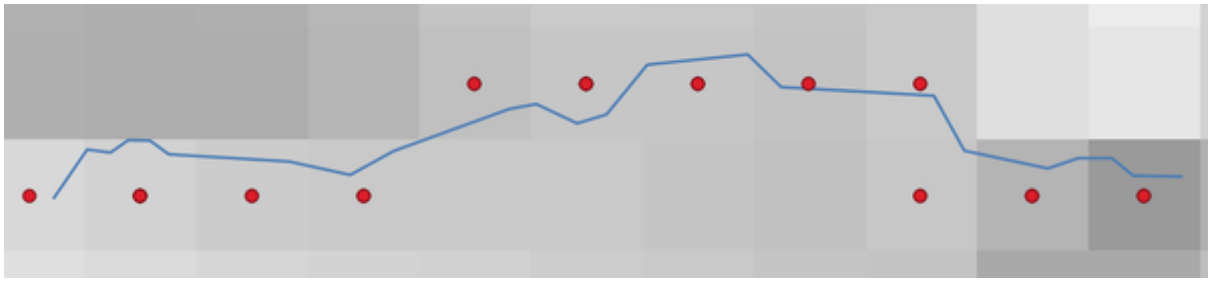
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Kurti taškus (pikselių centroidus) palei liniją

Kuria vektorinį taškų sluoksnį iš įvesties rastro ir linijų sluoksnio.

Taškai atitinka pikselių centroidus, kurie susikerta su linijų sluoksniu.



24.80 Fig. : Pikselių centroidų taškai

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Rastro sluoksnis	`INPUT_RASTER`	[rastras]	Įvesties rastro sluoksnis
Vektorinis sluoksnis	`INPUT_VECTOR`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Taškai palei liniją	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Gaunamas taškų sluoksnis su pikselių centroidais. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Taškai palei liniją	`OUTPUT`	[vektorius: taškas]	Gaunamas taškų sluoksnis su pikselių centroidais

Pythono kodas

Algoritmo ID: `qgis:generatepointspixelcentroidsalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

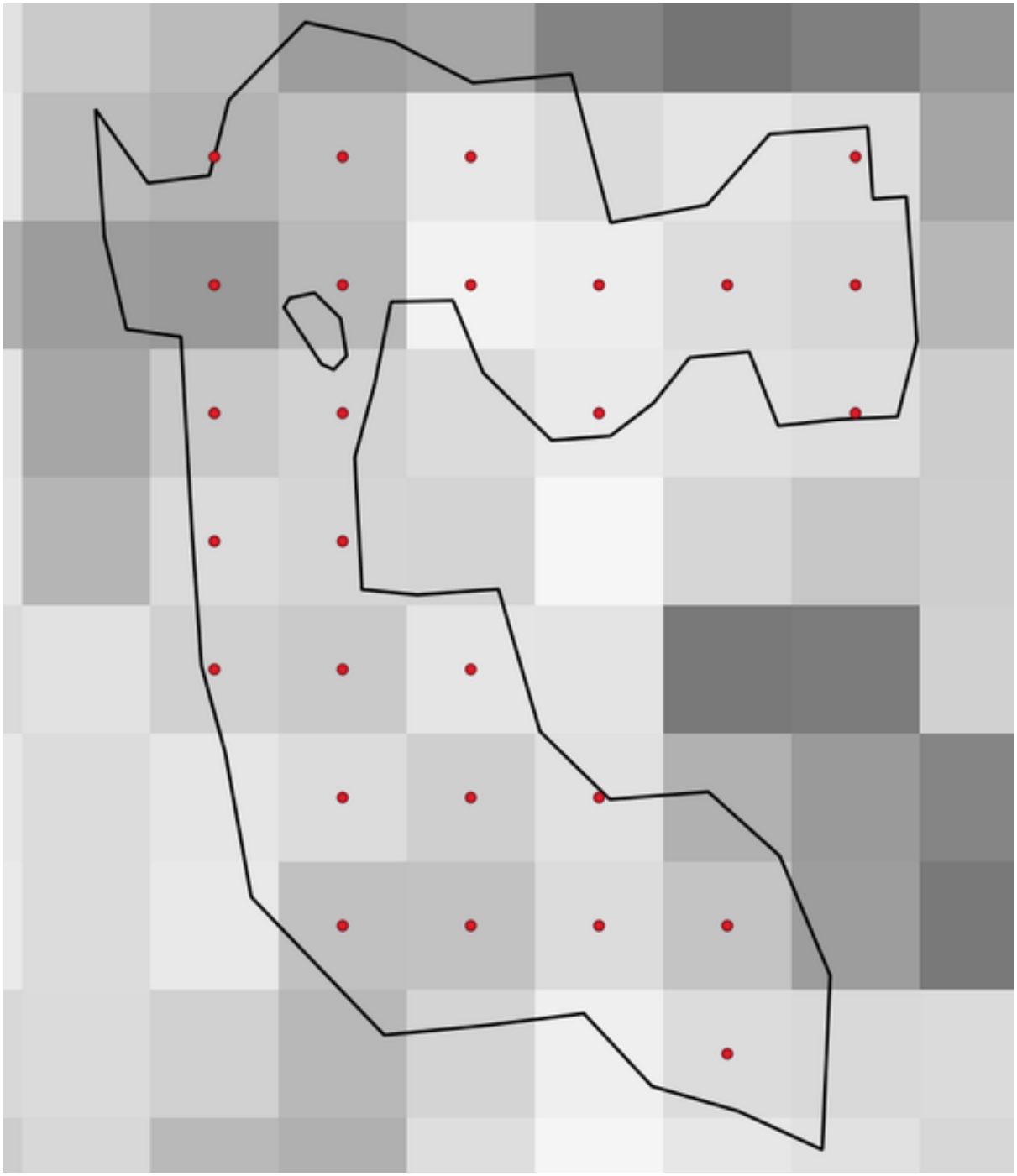
Generate points (pixel centroids) inside polygon

Generates a point vector layer from an input raster and polygon layer.

The points correspond to the pixel centroids that intersect the polygon layer.

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.



24.81 Fig. : Pikselių centroidų taškai

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Rastro sluoksnis	`INPUT_RASTER`	[rastras]	Įvesties rastro sluoksnis
Vektorinis sluoksnis	`INPUT_VECTOR`	[vektorius: poligonas]	Įvesties poligonų vektorinis sluoksnis
Points inside polygons	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Resulting point layer of pixel centroids. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Points inside polygons	`OUTPUT`	[vektorius: taškas]	Resulting point layer of pixel centroids

Pythono kodas

Algorithm ID: `native:generatepointspixelcentroidsinsidepolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Import geotagged photos

Creates a point layer corresponding to the geotagged locations from JPEG or HEIC/HEIF images from a source folder.

The point layer will contain a single PointZ feature per input file from which the geotags could be read. Any altitude information from the geotags will be used to set the point's Z value.

Besides longitude and latitude also altitude, direction and timestamp information, if present in the photo, will be added to the point as attributes.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input folder	`FOLDER`	[aplankas]	Path to the source folder containing the geotagged photos
Scan recursively	`RECURSIVE`	[boolean] Numatytas: ne	If checked, the folder and its subfolders will be scanned
Photos Pasirinktinis	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the point vector layer for the geotagged photos. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Invalid photos table Pasirinktinis	`INVALID`	[vektorius: lentelė] Numatytas: `[Praleisti išvestį]`	Specify the table of unreadable or non-geotagged photos. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Photos	`OUTPUT`	[vektorius: taškas]	Point vector layer with geotagged photos. The form of the layer is automatically filled with paths and photo previews settings.
Invalid photos table Pasirinktinis	`INVALID`	[vektorius: lentelė]	Table of unreadable or non-geotagged photos can also be created.

Pythono kodas

Algorithm ID: `native:importphotos`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Points to path

Converts a point layer to a line layer, by joining points in an order defined by an expression or a field in the input point layer.

Points can be grouped by a field or an expression to distinguish line features.

In addition to the line vector layer, a text file is output that describes the resulting line as a start point and a sequence of bearings / directions (relative to azimuth) and distances.

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input point layer	`INPUT`	[vektorius: taškas]	Input point vector layer
Create closed paths	`CLOSE_PATH`	[boolean] Numatytas: ne	If checked, the first and last points of the line will be connected and close the generated path
Order expression Pasirinktinis	`ORDER_EXPRESSI`	[išraiška]	Field or expression providing the order to connect the points in the path. If not set, the feature ID (`$id`) is used.
Sort text containing numbers naturally	`NATURAL_SORT`	[boolean] Numatytas: ne	If checked, naturally sorts the features based on the provided expression (i.e., ‚a9' < ‚a10').
Path group expression Pasirinktinis	`GROUP_EXPRESSI`	[išraiška]	Point features of the same value in the field or expression will be grouped in the same line. If not set, a single path is drawn with all the input points.
Paths	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the line vector layer of the path. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Directory for text output Pasirinktinis	`OUTPUT_TEXT_DI`	[aplankas] Numatytas: `[Praleisti išvestį]`	Specify the directory that will contain the description files of points and paths. *One of*: • Praleisti išvestį • Įrašyti į laikiną aplanką • Įrašyti į aplanką

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Paths	`OUTPUT`	[vektorius: linija]	Line vector layer of the path
Directory for text output	`OUTPUT_TEXT_DI`	[aplankas]	Directory containing description files of points and paths

Pythono kodas

Algorithm ID: `native:pointstopath`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

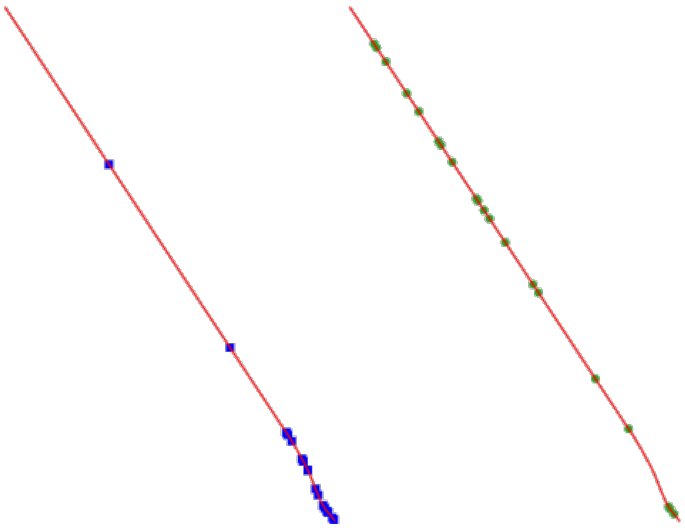
Random points along line

Creates a new point layer, with points placed on the lines of another layer.

For each line in the input layer, a given number of points is added to the resulting layer. The procedure for adding a point is to:

1. randomly select a line feature from the input layer
2. if the feature is multi-part, randomly select a part of it
3. randomly select a segment of that line
4. randomly select a position on that segment.

The procedure means that curved parts of the lines (with relatively short segments) will get more points than straight parts (with relatively long segments), as demonstrated in the illustration below, where the output of the *Random points along lines* algorithm can be compared with the output of the *Random points on lines* algorithm (that produces points with an, on average, even distribution along the lines).



24.82 Fig. : Example algorithm output. Left: *Random points along line*, right: *Random points on lines*

A minimum distance can be specified, to avoid points being too close to each other.

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Random points on lines

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input point layer	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Number of points	`POINTS_NUMBER`	[skaičius: sveikas skaičius] Numatytas: 1	Number of points to create
Minimum distance between points	`MIN_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	The minimum distance between points
Random points	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	The output random points. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Random points	`OUTPUT`	[vektorius: taškas]	The output random points layer.

Pythono kodas

Algorithm ID: `qgis:qgisrandompointsalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Random points in extent

Creates a new point layer with a given number of random points, all of them within a given extent.

A distance factor can be specified, to avoid points being too close to each other. If the minimum distance between points makes it impossible to create new points, either distance can be decreased or the maximum number of attempts may be increased.

Numatytas meniu: *Vektorius* ► *Tyrimo įrankiai*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input extent	`EXTENT`	[apimtis]	Map extent for the random points Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Number of points	`POINTS_NUMBER`	[skaičius: sveikas skaičius] Numatytas: 1	Number of point to create
Minimum distance between points	`MIN_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	The minimum distance between points
Paskirties CRS	`TARGET_CRS`	[crs] Numatyta: *Projekto CRS*	CRS of the random points layer
Random points	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	The output random points. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Maximum number of search attempts given the minimum distance	`MAX_ATTEMPTS`	[skaičius: sveikas skaičius] Default: 200	Maximum number of attempts to place the points

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Random points	`OUTPUT`	[vektorius: taškas]	The output random points layer.

Pythono kodas

Algorithm ID: `native:randompointsinextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Random points in layer bounds

Creates a new point layer with a given number of random points, all of them within the extent of a given layer.

A minimum distance can be specified, to avoid points being too close to each other.

Numatytas meniu: *Vektorius* ► *Tyrimo įrankiai*

> **Įspėjimas**
>
> Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Input polygon layer defining the area
Number of points	`POINTS_NUMBER`	[skaičius: sveikas skaičius] Numatytas: 1	Number of points to create
Minimum distance between points	`MIN_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	The minimum distance between points

continues on next page

Table 24.520 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Random points	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	The output random points. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Random points	`OUTPUT`	[vektorius: taškas]	The output random points layer.

Pythono kodas

Algorithm ID: `qgis:randompointsinlayerbounds`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

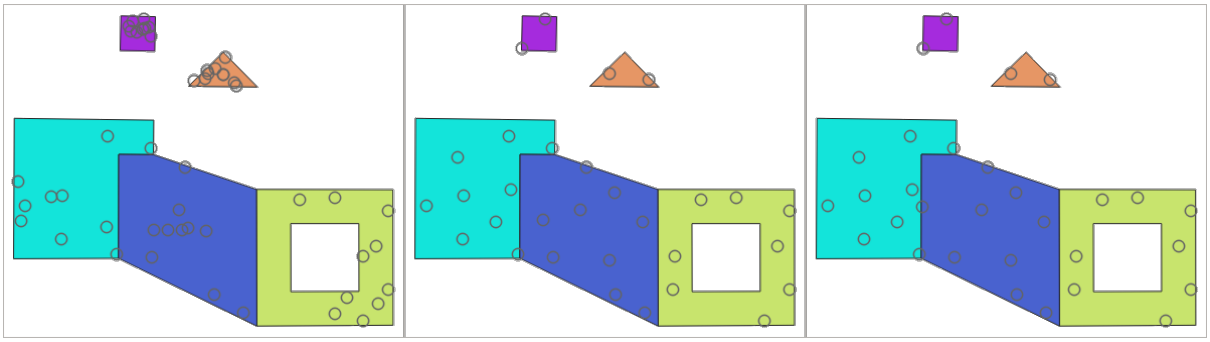
Random points in polygons

Creates a point layer with points placed inside the polygons of another layer.

For each feature (polygon / multi-polygon) geometry in the input layer, the given number of points is added to the result layer.

Per feature and global minimum distances can be specified in order to avoid points being too close in the output point layer. If a minimum distance is specified, it may not be possible to generate the specified number of points for each feature. The total number of generated points and missed points are available as output from the algorithm.

The illustration below shows the effect of per feature and global minimum distances and zero/non-zero minimum distances (generated with the same seed, so at least the first point generated will be the same).



24.83 Fig. : Ten points per polygon feature, *left*: min. distances = 0, *middle*: min.distances = 1, *right*: min. distance = 1, global min. distance = 0

The maximum number of tries per point can be specified. This is only relevant for non-zero minimum distance.

A seed for the random number generator can be provided, making it possible to get identical random number sequences for different runs of the algorithm.

The attributes of the polygon feature on which a point was generated can be included (*Include polygon attributes*).

If you want approximately the same point density for all the features, you can data-define the number of points using the area of the polygon feature geometry.

Taip pat žiūrėkite

Random points inside polygons

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input polygon layer	`INPUT`	[vektorius: linija]	Įvesties poligonų vektorinis sluoksnis
Number of points for each feature	`POINTS_NUMBER`	[skaičius: sveikas skaičius] Numatytas: 1	Number of points to create
Minimum distance between points Pasirinktinis	`MIN_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	The minimum distance between points within one polygon feature
Random points in polygons	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	The output random points. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Global minimum distance between points Pasirinktinis	`MIN_DISTANCE_G`	[skaičius: slankaus kablelio] Numatytas: 0.0	The global minimum distance between points. Should be smaller than the *Minimum distance between points (per feature)* for that parameter to have an effect.
Maximum number of search attempts (for Min. dist. > 0) Pasirinktinis	`MAX_TRIES_PER_`	[skaičius: sveikas skaičius] Numatytas: 10	The maximum number of tries per point. Only relevant if the minimum distance between points is set (and greater than 0).
Random seed Pasirinktinis	`SEED`	[skaičius: sveikas skaičius] Numatytas: jokio	The seed to use for the random number generator.

continues on next page

Table 24.523 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Include polygon attributes	`INCLUDE_POLYGO`	[boolean] Numatytas: Taip	If set, a point will get the attributes from the line on which it is placed.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Random points in polygons	`OUTPUT`	[vektorius: taškas]	The output random points layer.
Number of features with empty or no geometry	`FEATURES_WITH_`	[skaičius: sveikas skaičius]	
Total number of points generated	`OUTPUT_POINTS`	[skaičius: sveikas skaičius]	
Number of missed points	`POINTS_MISSED`	[skaičius: sveikas skaičius]	The number of points that could not be generated due to the minimum distance constraint.
Number of features with missed points	`POLYGONS_WITH_`	[skaičius: sveikas skaičius]	Not including features with empty or no geometry

Pythono kodas

Algorithm ID: `native:randompointsinpolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Random points inside polygons

Creates a new point layer with a given number of random points inside each polygon of the input polygon layer.

Two sampling strategies are available:

- Points count: number of points for each feature
- Points density: density of points for each feature

A minimum distance can be specified, to avoid points being too close to each other.

Numatytas meniu: *Vektorius* ► *Tyrimo įrankiai*

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Random points in polygons

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Įvesties poligonų vektorinis sluoksnis
Sampling strategy	`STRATEGY`	[sąrašas] Numatytas: 0	Sampling strategy to use. One of: • 0 — Points count: number of points for each feature • 1 — Points density: density of points for each feature
Point count or density	`VALUE`	[skaičius: slankaus kablelio] Numatytas: 1.0	The number or density of points, depending on the chosen *Sampling strategy*.
Minimum distance between points	`MIN_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	The minimum distance between points
Random points	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	The output random points. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Random points	`OUTPUT`	[vektorius: taškas]	The output random points layer.

Pythono kodas

Algorithm ID: `qgis:randompointsinsidepolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Random points on lines

Creates a point layer with points placed on the lines of another layer.

For each feature (line / multi-line) geometry in the input layer, the given number of points is added to the result layer.

Per feature and global minimum distances can be specified in order to avoid points being too close in the output point layer. If a minimum distance is specified, it may not be possible to generate the specified number of points for each feature. The total number of generated points and missed points are available as output from the algorithm.

The illustration below shows the effect of per feature and global minimum distances and zero/non-zero minimum distances (generated with the same seed, so at least the first point generated will be the same).

24.84 Fig. : Five points per line feature, *left*: min. distances = 0, *middle*: min.distances != 0, *right*: min. distance != 0, global min. distance = 0

The maximum number of tries per point can be specified. This is only relevant for non-zero minimum distance.

A seed for the random number generator can be provided, making it possible to get identical random number sequences for different runs of the algorithm.

The attributes of the line feature on which a point was generated can be included (*Include line attributes*).

If you want approximately the same point density for all the line features, you can data-define the number of points using the length of the line feature geometry.

Taip pat žiūrėkite

Random points along line

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties linijų sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Number of points for each feature	`POINTS_NUMBER`	[skaičius: sveikas skaičius] Numatytas: 1	Number of points to create
Minimum distance between points (per feature) Pasirinktinis	`MIN_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	The minimum distance between points within one line feature
Random points on lines	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	The output random points. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Global minimum distance between points Pasirinktinis	`MIN_DISTANCE_G`	[skaičius: slankaus kablelio] Numatytas: 0.0	The global minimum distance between points. Should be smaller than the *Minimum distance between points (per feature)* for that parameter to have an effect.
Maximum number of search attempts (for Min. dist. > 0) Pasirinktinis	`MAX_TRIES_PER_`	[skaičius: sveikas skaičius] Numatytas: 10	The maximum number of tries per point. Only relevant if the minimum distance between points is set (and greater than 0).
Random seed Pasirinktinis	`SEED`	[skaičius: sveikas skaičius] Numatytas: jokio	The seed to use for the random number generator.
Include line attributes	`INCLUDE_LINE_A`	[boolean] Numatytas: Taip	If set, a point will get the attributes from the line on which it is placed.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Random points on lines	`OUTPUT`	[vektorius: taškas]	The output random points layer.
Number of features with empty or no geometry	`FEATURES_WITH_`	[skaičius: sveikas skaičius]	
Number of features with missed points	`LINES_WITH_MIS`	[skaičius: sveikas skaičius]	Not including features with empty or no geometry
Total number of points generated	`POINTS_GENERAT`	[skaičius: sveikas skaičius]	
Number of missed points	`POINTS_MISSED`	[skaičius: sveikas skaičius]	The number of points that could not be generated due to the minimum distance constraint.

Pythono kodas

Algorithm ID: `native:randompointsonlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster pixels to points

Creates a vector layer of points corresponding to each pixel in a raster layer.

Converts a raster layer to a vector layer, by creating point features for each individual pixel's center in the raster layer. Any NoData pixels are skipped in the output.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Rastro sluoksnis	`INPUT_RASTER`	[rastras]	Įvesties rastro sluoksnis
Juostos numeris	`RASTER_BAND`	[rastro juosta]	Raster band to extract data from
Lauko pavadinimas	`FIELD_NAME`	[tekstas] Default: ‚VALUE‘	Name of the field to store the raster band value
Vector points	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the resulting point layer of pixels centroids. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Vector points	`OUTPUT`	[vektorius: taškas]	Resulting point layer with pixels centroids

Pythono kodas

Algorithm ID: `native:pixelstopoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster pixels to polygons

Creates a vector layer of polygons corresponding to each pixel in a raster layer.

Converts a raster layer to a vector layer, by creating polygon features for each individual pixel's extent in the raster layer. Any NoData pixels are skipped in the output.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Rastro sluoksnis	`INPUT_RASTER`	[rastras]	Įvesties rastro sluoksnis
Juostos numeris	`RASTER_BAND`	[rastro juosta]	Raster band to extract data from
Lauko pavadinimas	`FIELD_NAME`	[tekstas] Default: ‚VALUE‘	Name of the field to store the raster band value
Vector polygons	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the resulting polygon layer of pixel extents. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Vector polygons	`OUTPUT`	[vektorius: poligonas]	Resulting polygon layer of pixel extents

Pythono kodas

Algorithm ID: `native:pixelstopolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Regular points

Creates a new point layer with its points placed in a regular grid within a given extent.

The grid is specified either by the spacing between the points (same spacing for all dimensions) or by the number of points to generate. In the latter case, the spacing will be determined from the extent. In order to generate a full rectangular grid, at least the number of points specified by the user is generated for the latter case.

Random offsets to the point spacing can be applied, resulting in a non-regular point pattern.

Numatytas meniu: *Vektorius* ► *Tyrimo įrankiai*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input extent (xmin, xmax, ymin, ymax)	`EXTENT`	[apimtis]	Map extent for the random points Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Point spacing/count	`SPACING`	[skaičius: sveikas skaičius] Numatytas: 100	Spacing between the points, or the number of points, depending on whether `Use point spacing` is checked or not.
Initial inset from corner (LH side)	`INSET`	[skaičius: slankaus kablelio] Numatytas: 0.0	Offsets the points relative to the upper left corner. The value is used for both the X and Y axis.
Apply random offset to point spacing	`RANDOMIZE`	[boolean] Numatytas: ne	If checked the points will have a random spacing
Use point spacing	`IS_SPACING`	[boolean] Numatytas: Taip	If unchecked the point spacing is not taken into account

continues on next page

Table 24.534 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Output layer CRS	`CRS`	[crs] Numatyta: *Projekto CRS*	CRS of the random points layer
Regular points	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output regular point layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Regular points	`OUTPUT`	[vektorius: taškas]	The output regular point layer.

Pythono kodas

Algorithm ID: `qgis:regularpoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.25 Bendri vektoriniai

Priskirti projekciją

Priskiria vektoriniam sluoksniui naują projekciją.

Sukuria naują sluoksnį su lygiai tokiais pačiais geoobjektais ir geometrijomis kaip ir įvesties, tik su priskirta nauja CRS. Geometrijos **nėra** perprojektuojamos, joms tiesiog priskiriama kita CRS.

Šį algoritmą galima naudoti taisant sluoksnius, kuriems buvo priskirta neteisinga projekcija.

Šis algoritmas nekeičia atributų.

Taip pat žiūrėkite

Define projection, *Find projection*, *Reproject layer*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Vektorinis sluoksnis su neteisinga ar trūkstama CRS
Priskiriama CRS	`CRS`	[crs] Numatyta: `EPSG:4326 - WGS84`	Parinkite naują CRS, kurią reikia priskirti vektoriniam sluoksniui
Priskiriama CRS Pasirinktinis	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Priskiriama CRS	`OUTPUT`	[toks pat, kaip įvestis]	Vektorinis sluoksnis su priskirta projekcija

Pythono kodas

Algoritmo ID: `native:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Paketinis Nominatim geokodavimas

Vykdo paketinį įvesties sluoksnio tekstinio lauko geokodavimą naudojant Nominatim paslaugą. Išvesties sluoksnis turės taško geometriją, atspindinčią geokoduotą vietą bei kelis atributus, susijusius su geokoduota vieta.

Leidžia taškų *geoobjektus keisti vietoje*

Pastaba

Šis algoritmas atitinka Nominatim geokodavimo paslaugos, kurią teikia OpenStreetMap Foundation, naudojimo politiką.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Vektorinis sluoksnis su geokoduojamais objektais
Adreso laukas	`FIELD`	[lentelės laukas: tekstas]	Laukas su geokoduojamu adresu
Geokoduotas	`OUTPUT`	[vektorius: taškai] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties sluoksnį, kuriame yra tik geokoduoti adresai. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Geokoduotas	`OUTPUT`	[vektorius: taškai]	Vektorinis sluoksnis su taškiniais geoobjektais, atitinkančiais geokoduotus adresus

Pythono kodas

Algoritmo ID: `native:batchnominatimgeocoder`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Keisti sluoksnį į erdvines žymeles

Kuria erdvines žymeles, atitinkančias sluoksnyje esančių geoobjektų apimtis.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija, poligonas]	Įvesties vektorinis sluoksnis
Žymelių paskirtis	`DESTINATION`	[sąrašas] Numatytas: 0	Parinkite žymelių paskirtį. Vienas iš: • 0 — Projekto žymelės • 1 — Naudotojo žymelės

continues on next page

Table 24.540 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Pavadinimo laukas	`NAME_EXPRESSIO`	[išraiška]	Laukas arba išraiška, kuri suteiks sukurtoms žymelėms pavadinimą
Grupės laukas	`GROUP_EXPRESSI`	[išraiška]	Laukas ar išraiška, kuri suteiks sukurtoms žymelėms grupes

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Pridėti žymelių skaičius	`COUNT`	[skaičius: sveikas skaičius]	

Pythono kodas

Algoritmo ID: `native:layertobookmarks`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Keisti erdvines žymeles į sluoksnį

Kuria naują sluoksnį, kuriame yra poligono geoobjektai su įrašytomis erdvinėmis žymelėmis. Eksportą galima apriboti tik dabartinio projekto žymelėmis, visomis naudotojo žymelėmis arba abiem.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Žymelių šaltinis	`SOURCE`	[enumeration] [list] Numatytas: [0,1]	Parinktie žymelių šaltinį(ius). Vienas ar daugiau iš: • 0 — Projekto žymelės • 1 — Naudotojo žymelės
Išvesties CRS	`CRS`	[crs] Numatyta: `EPSG:4326 - WGS 84`	Išvesties sluoksnio CRS
Išvestis	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvestis	`OUTPUT`	[vektorius: poligonas]	Išvesties (žymelių) vektorinis sluoksnis

Pythono kodas

Algoritmo ID: `native:bookmarkstolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Create attribute index

Creates an index against a field of the attribute table to speed up queries. The support for index creation depends on both the layer's data provider and the field type.

No outputs are created: the index is stored on the layer itself.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Select the vector layer you want to create an attribute index for
Attribute to index	`FIELD`	[tablefield: any]	Field of the vector layer

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Indexed layer	`OUTPUT`	[toks pat, kaip įvestis]	A copy of the input vector layer with an index for the specified field

Pythono kodas

Algorithm ID: `native:createattributeindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Create spatial index

Creates an index to speed up access to the features in a layer based on their spatial location. Support for spatial index creation is dependent on the layer's data provider.

No new output layers are created.

Default menu: *Vector ► Data Management Tools*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Indexed layer	`OUTPUT`	[toks pat, kaip įvestis]	A copy of the input vector layer with a spatial index

Pythono kodas

Algorithm ID: `native:createspatialindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Define projection

Sets an existing layer's projection to the provided CRS without reprojecting features.

Contrary to the *Priskirti projekciją* algorithm, it modifies the current layer and will not output a new layer.

Pastaba

For Shapefile datasets, the `.prj` and `.qpj` files will be overwritten - or created if missing - to match the provided CRS.

Default menu: *Vector ► Data Management Tools*

Taip pat žiūrėkite

Priskirti projekciją, *Find projection*, *Reproject layer*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Vector layer with missing projection information
CRS	`CRS`	[crs]	Select the CRS to assign to the vector layer

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
	`INPUT`	[toks pat, kaip įvestis]	The input vector layer with the defined projection

Pythono kodas

Algorithm ID: `native:definecurrentprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Delete duplicate geometries

Finds and removes duplicated geometries.

Attributes are not checked, so in case two features have identical geometries but different attributes, only one of them will be added to the result layer.

Pastaba

This algorithm does not require valid geometries as input.

Taip pat žiūrėkite

Drop geometries, *Remove null geometries*, *Delete duplicates by attribute*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	The layer with duplicate geometries you want to clean
Cleaned	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output layer containing the unique features. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Count of discarded duplicate records	`DUPLICATE_COUN`	[skaičius: sveikas skaičius]	Count of discarded duplicate records
Cleaned	`OUTPUT`	[toks pat, kaip įvestis]	The output layer without any duplicated geometries
Count of retained records	`RETAINED_COUNT`	[skaičius: sveikas skaičius]	Count of unique records

Pythono kodas

Algorithm ID: `native:deleteduplicategeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Delete duplicates by attribute

Deletes duplicate rows by only considering the specified field(s). The first matching row will be retained, and duplicates will be discarded.

Optionally, these duplicate records can be saved to a separate output for analysis.

Taip pat žiūrėkite

Delete duplicate geometries

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties sluoksnis
Fields to match duplicates by	`FIELDS`	[lentelės laukas: bet koks] [sąrašas]	Fields defining duplicates. Features with identical values for all these fields are considered duplicates.
Filtered (no duplicates)	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output layer containing the unique features. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Filtered (duplicates) Pasirinktinis	`DUPLICATES`	[toks pat, kaip įvestis] Numatytas: `[Praleisti išvestį]`	Nurodykite išvesties sluoksnį, kuriame bus tik dublikatai. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Filtered (duplicates)	`DUPLICATES`	[toks pat, kaip įvestis]	Vector layer containing the duplicate features.
Count of discarded duplicate records	`DUPLICATE_COUN`	[skaičius: sveikas skaičius]	Count of discarded duplicate records
Filtered (no duplicates)	`OUTPUT`	[toks pat, kaip įvestis]	Vector layer containing the unique features.
Count of retained records	`RETAINED_COUNT`	[skaičius: sveikas skaičius]	Count of unique records

Pythono kodas

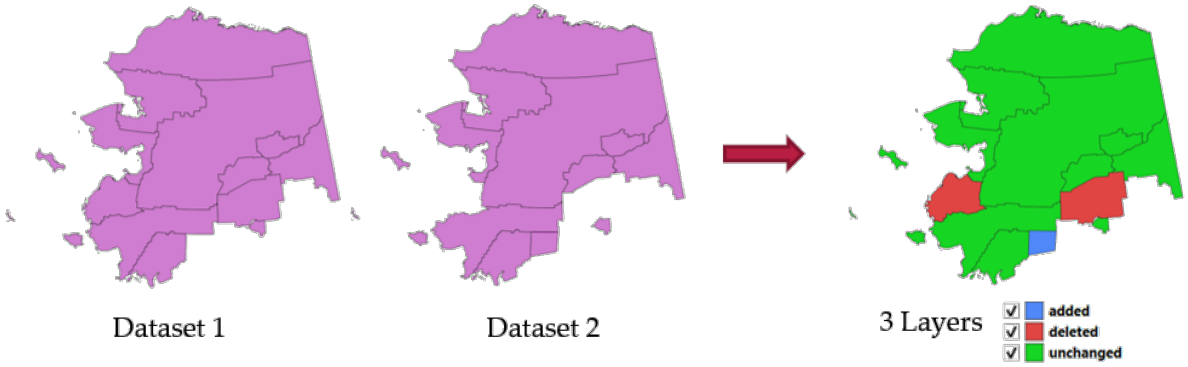
Algorithm ID: `native:removeduplicatesbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Detect dataset changes

Compares two vector layers, and determines which features are unchanged, added or deleted between the two. It is designed for comparing two different versions of the same dataset.



24.85 Fig. : Detect dataset change example

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Original layer	`ORIGINAL`	[vektorius: geometrija]	The vector layer considered as the original version
Revised layer	`REVISED`	[vektorius: geometrija]	The revised or modified vector layer
Attributes to consider for match Pasirinktinis	`COMPARE_ATTRIB`	[lentelės laukas: bet koks] [sąrašas]	Attributes to consider for match. By default, all attributes are compared.
Geometry comparison behavior Pasirinktinis	`MATCH_TYPE`	[sąrašas] Numatytas: 1	Defines the criteria for comparison. Options: • 0 — Exact Match: includes the order and vertices count of geometries • 1 — Tolerant Match (Topological Equality): geometries are considered equal
Unchanged features Pasirinktinis	`UNCHANGED`	[vector: same as Original layer]	Specify the output vector layer containing the unchanged features. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

continues on next page

Table 24.554 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Added features Pasirinktinis	`ADDED`	[vector: same as Original layer]	Specify the output vector layer containing the added features. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Deleted features Pasirinktinis	`DELETED`	[vector: same as Original layer]	Specify the output vector layer containing the deleted features. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Unchanged features	`UNCHANGED`	[vector: same as Original layer]	Vector layer containing the unchanged features.
Added features	`ADDED`	[vector: same as Original layer]	Vector layer containing the added features.
Deleted features	`DELETED`	[vector: same as Original layer]	Vector layer containing the deleted features.
Count of unchanged features	`UNCHANGED_COUN`	[skaičius: sveikas skaičius]	Count of unchanged features.
Count of features added in revised layer	`ADDED_COUNT`	[skaičius: sveikas skaičius]	Count of features added in revised layer.
Count of features deleted from original layer	`DELETED_COUNT`	[skaičius: sveikas skaičius]	Count of features deleted from original layer.

Pythono kodas

Algorithm ID: `native:detectvectorchanges`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Drop geometries

Creates a simple *geometryless* copy of the input layer attribute table. It keeps the attribute table of the source layer.

If the file is saved in a local folder, you can choose between many file formats.

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Taip pat žiūrėkite

Delete duplicate geometries, *Remove null geometries*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Dropped geometries	`OUTPUT`	[vektorius: lentelė]	Specify the output geometryless layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Dropped geometries	`OUTPUT`	[vektorius: lentelė]	The output geometryless layer. A copy of the original attribute table.

Pythono kodas

Algorithm ID: `native:dropgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Vykdyti SQL

Runs a simple or complex query based only on SELECT with `SQL` syntax on the source layer.

Input datasources are identified with `input1`, `input2`... `inputN` and a simple query will look like `SELECT * FROM input1`.

Beside a simple query, you can add expressions or variables within the `SQL query` parameter itself. This is particulary useful if this algorithm is executed within a Processing model and you want to use a model input as a parameter of the query. An example of a query will then be `SELECT * FROM [% @table %]` where `@table` is the variable that identifies the model input.

The result of the query will be added as a new layer.

Taip pat žiūrėkite

SpatiaLite execute SQL, *PostgreSQL execute SQL*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Additional input datasources (called input1, .., inputN in the query)	`INPUT_DATASOUR`	[vektorius: bet koks] [sąrašas]	List of layers to query. In the SQL editor you can refer these layers with their **real** name or also with **input1**, **input2**, **inputN** depending on how many layers have been chosen.
SQL query	`INPUT_QUERY`	[tekstas]	Type the string of your SQL query, e.g. `SELECT * FROM input1`.
Unique identifier field Pasirinktinis	`INPUT_UID_FIEL`	[tekstas]	Specify the column with unique ID
Geometry field Pasirinktinis	`INPUT_GEOMETRY`	[tekstas]	Specify the geometry field
Geometry type Pasirinktinis	`INPUT_GEOMETRY`	[sąrašas] Numatytas: 0	Choose the geometry of the result. By default the algorithm will autodetect it. One of: • 0 — Autodetect • 1 — No geometry • 2 — Point • 3 — LineString • 4 — Polygon • 5 — MultiPoint • 6 — MultiLineString • 7 — MultiPolygon
CRS Pasirinktinis	`INPUT_GEOMETRY`	[crs]	The CRS to assign to the output layer
SQL Output	`OUTPUT`	[vektorius: geometrija] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output layer created by the query. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
SQL Output	`OUTPUT`	[vektorius: geometrija]	Pagal užklausą sukurtas vektorinis sluoksnis

Pythono kodas

Algorithm ID: `qgis:executesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export layers to DXF

Exports layers to DXF file. For each layer, you can choose a field whose values are used to split features in generated destination layers in DXF output.

Taip pat žiūrėkite

Creating new DXF files

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`LAYERS`	[vektorius: geometrija] [sąrašas]	List of input vector layers with options associated (filled as a `QgsProcessingParameterDxfLayers` item — done in GUI by pressing *Configure Layer…* button for each selected layer): **Layer [string] (`layer`)** Full path of the input layer to export **Output layer attribute [tablefield: any] (`attributeIndex`)** Attribute index to split the input layer using unique values **Output layer name [string] (`overriddenLayerName`)** Overridden layer name to be used in the exported DXF file (the original project layer remains untouched) **Allow data defined symbol blocks** [boolean] (`buildDataDefinedBlocks`) **Maximum number of symbol blocks [numeric: integer] (`dataDefinedBlocksMaximumNumberOfClasses`)** `-1` means no limitation.
Symbology mode	`SYMBOLOGY_MODE`	[sąrašas] Numatytas: 0	Type of symbology to apply to output layers. You can choose between: • 0 — No Symbology • 1 — Feature Symbology • 2 — Symbol Layer Symbology
Symbology scale	`SYMBOLOGY_SCAL`	[scale] Default: 1:1 000 000	Default scale of data export.
Map theme Pasirinktinis	`MAP_THEME`	[map theme]	Match layer styling to the provided map theme.
Koduotė	`ENCODING`	[sąrašas]	Encoding to apply to layers.
CRS	`CRS`	[crs]	Choose the CRS for the output layer.
Apimtis Pasirinktinis	`EXTENT`	[apimtis]	Limit exported features to those with geometries intersecting the provided extent.
Use layer title as name	`USE_LAYER_TITL`	[boolean] Numatytas: ne	Name the output layer with the layer title (as set in layer metadata or QGIS Server properties) instead of the layer name.
Force 2D	`FORCE_2D`	[boolean] Numatytas: ne	
Export labels as MTEXT elements	`MTEXT`	[boolean] Numatytas: Taip	Exports labels as MTEXT or TEXT elements
Use only selected features	`SELECTED_FEATU`	[boolean] Numatytas: ne	Exports only the selected features.
DXF	`OUTPUT`	[file] Numatytas: `[Įrašyti į laikiną failą]`	Specification of the output DXF file. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
DXF	`OUTPUT`	[file]	`.DXF` file containing the input layers

Pythono kodas

Algorithm ID: `native:dxfexport`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Extract selected features

Saves the selected features as a new layer.

> **Pastaba**
>
> If the selected layer has no selected features, the newly created layer will be empty.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Layer to save the selection from
Selected features	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the vector layer for the selected features. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Selected features	`OUTPUT`	[toks pat, kaip įvestis]	Vector layer with only the selected features, or no feature if none was selected.

Pythono kodas

Algorithm ID: `native:saveselectedfeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Extract Shapefile encoding

Extracts the attribute encoding information embedded in a Shapefile. Both the encoding specified by an optional `.cpg` file and any encoding details present in the `.dbf` LDID header block are considered.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	ESRI Shapefile (`.SHP`) Layer to extract the encoding information.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Shapefile encoding	`ENCODING`	[tekstas]	Encoding information specified in the input file
CPG encoding	`CPG_ENCODING`	[tekstas]	Encoding information specified in any optional `.CPG` file
LDID encoding	`LDID_ENCODING`	[tekstas]	Encoding information specified in `.dbf` LDID header block

Pythono kodas

Algorithm ID: `native:shpencodinginfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Find projection

Creates a shortlist of candidate coordinate reference systems, for instance for a layer with an unknown projection.

The area that the layer is expected to cover must be specified via the target area parameter. The coordinate reference system for this target area must be known to QGIS.

The algorithm operates by testing the layer's extent in every known reference system and then listing any for which the bounds would be near the target area if the layer was in this projection.

Taip pat žiūrėkite

Priskirti projekciją, *Define projection*, *Reproject layer*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Layer with unknown projection
Target area for layer (xmin, xmax, ymin, ymax)	`TARGET_AREA`	[apimtis]	The area that the layer covers. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
CRS candidates	`OUTPUT`	[vektorius: lentelė] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the table (geometryless layer) for the CRS suggestions (EPSG codes). *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
CRS candidates	`OUTPUT`	[vektorius: lentelė]	A table with all the CRS (EPSG codes) of the matching criteria.

Pythono kodas

Algorithm ID: `native:findprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

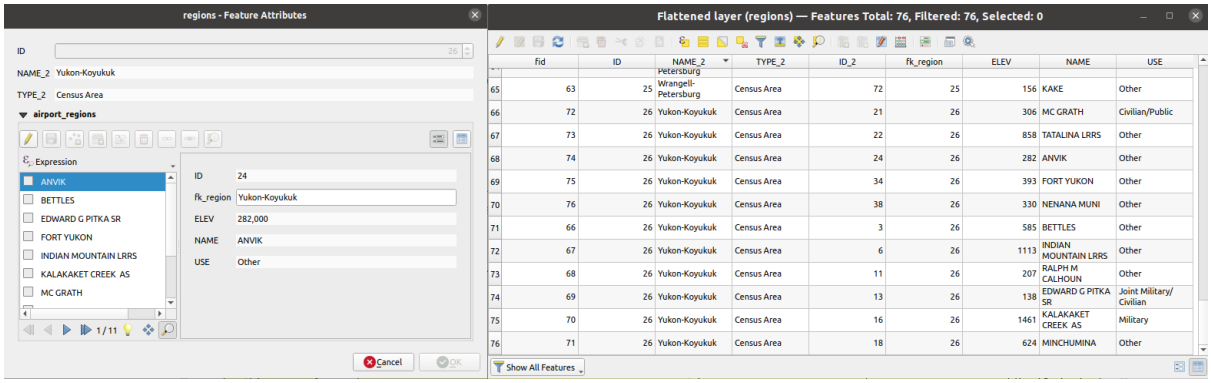
algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Flatten relationship

Flattens a *relationship* for a vector layer, exporting a single layer containing one parent feature per related child feature. This master feature contains all the attributes for the related features. This allows to have the relation as a plain table that can be e.g. exported to CSV.

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.



24.86 Fig. : Form of a region with related children (left) - A duplicate region feature for each related child, with joined attributes (right)

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Layer with the relationship that should be de-normalized
Flattened Layer Pasirinktinis	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output (flattened) layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Flattened layer	`OUTPUT`	[toks pat, kaip įvestis]	A layer containing master features with all the attributes for the related features

Pythono kodas

Algorithm ID: `native:flattenrelationships`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Join attributes by field value

Takes an input vector layer and creates a new vector layer that is an extended version of the input one, with additional attributes in its attribute table.

The additional attributes and their values are taken from a second vector layer. An attribute is selected in each of them to define the join criteria.

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Join attributes by nearest, *Join attributes by location*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Input vector layer. The output layer will consist of the features of this layer with attributes from matching features in the second layer.
Table field	`FIELD`	[tablefield: any]	Field of the source layer to use for the join
Input layer 2	`INPUT_2`	[vektorius: bet koks]	Layer with the attribute table to join
Table field 2	`FIELD_2`	[tablefield: any]	Field of the second (join) layer to use for the join The type of the field must be equal to (or compatible with) the input table field type.
Layer 2 fields to copy Pasirinktinis	`FIELDS_TO_COPY`	[lentelės laukas: bet koks] [sąrašas]	Select the specific fields you want to add. By default all the fields are added.
Join type	`METHOD`	[sąrašas] Numatytas: 1	The type of the final joined layer. One of: • 0 — Create separate feature for each matching feature (one-to-many) • 1 — Take attributes of the first matching feature only (one-to-one)
Discard records which could not be joined	`DISCARD_NONMAT`	[boolean] Numatytas: Taip	Check if you don't want to keep the features that could not be joined
Joined field prefix Pasirinktinis	`PREFIX`	[tekstas]	Add a prefix to joined fields in order to easily identify them and avoid field name collision
Joined layer Pasirinktinis	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer for the join. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Unjoinable features from first layer Pasirinktinis	`NON_MATCHING`	[toks pat, kaip įvestis] Numatytas: `[Praleisti išvestį]`	Specify the output vector layer for unjoinable features from first layer. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Number of joined features from input table	`JOINED_COUNT`	[skaičius: sveikas skaičius]	
Unjoinable features from first layer Pasirinktinis	`NON_MATCHING`	[toks pat, kaip įvestis]	Vector layer with the non-matched features
Joined layer Pasirinktinis	`OUTPUT`	[toks pat, kaip įvestis]	Output vector layer with added attributes from the join
Number of unjoinable features from input table Pasirinktinis	`UNJOINABLE_COU`	[skaičius: sveikas skaičius]	

Pythono kodas

Algorithm ID: `native:joinattributestable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Join attributes by location

Takes an input vector layer and creates a new vector layer that is an extended version of the input one, with additional attributes in its attribute table.

The additional attributes and their values are taken from a second vector layer. A spatial criteria is applied to select the values from the second layer that are added to each feature from the first layer.

Default menu: *Vector ► Data Management Tools*

Įspėjimas

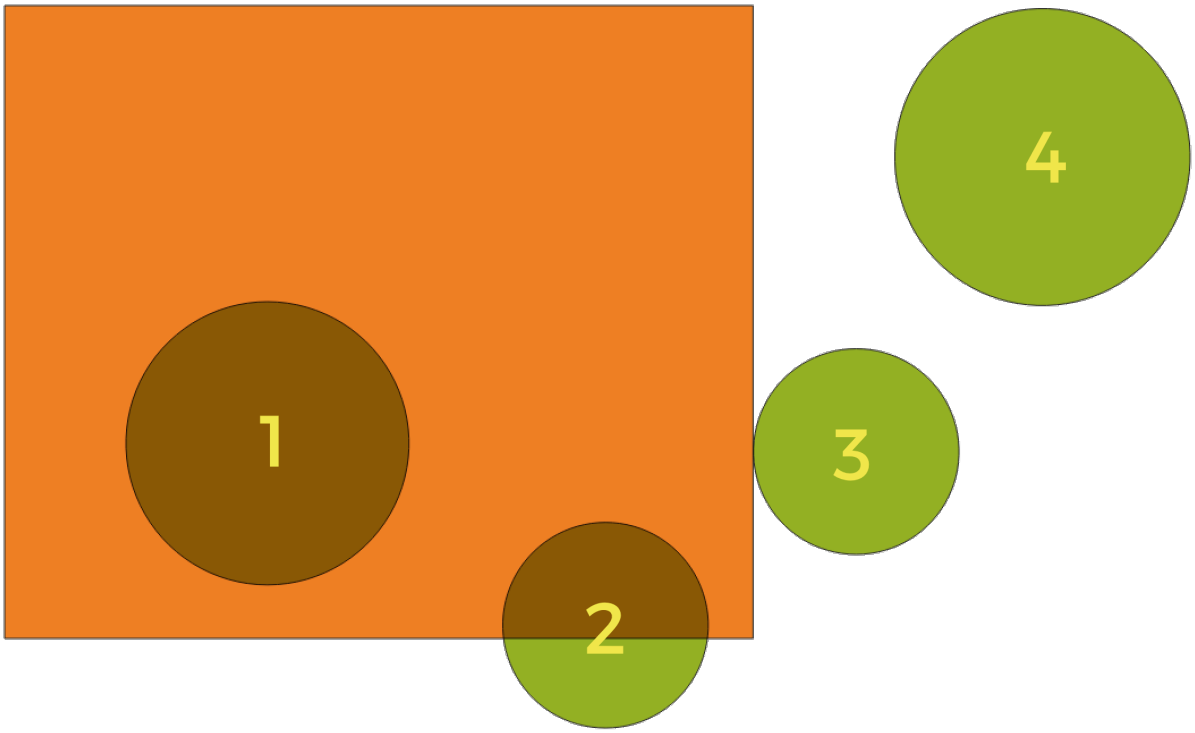
Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Join attributes by nearest, *Join attributes by field value*, *Join attributes by location (summary)*

Erdvinių ryšių tyrinėjimas

Geometriniai predikatai yra funkcijos, kurios naudojamos nustatant, kokį erdvinį ryšį geoobjektas turi su kitu, lyginant ar ir kaip jų geometrijos dalinasi erdvės dalimi.



24.87 Fig. : Erdvinių ryšių tarp sluoksnių paieška

Naudojant aukščiau pateiktą iliustraciją, mes ieškome žalių apskritimų, erdviškai lygindami juos su oranžiniu stačiakampiu geoobjektus. Galimi geometriniai predikatai yra:

Susikerta
: Tikrina, ar viena geometrija kerta kitą. Grąžina 1 (true), jei geometrijos erdviškai kertasi (dalinasi kažkokia erdvės dalimi - persidengia arba liečia) ir 0, jei to nėra. Iliustracijoje aukščiau True bus grąžinama apskritimams 1, 2 ir 3.

Įtraukia
: Grąžina 1 (true) tada ir tik tada, jei nei vienas b taškas nėra už a ribų, ir bent vienas b vidaus taškas yra a viduje. Iliustracijoje nei vienas apskritimas nebūtų grąžintas, bet stačiakampis būtų, jei jūs ieškotumėte atvirkščiai, nes pilnai stačiakampyje yra 1 apskritimas. Tai yra priešingybė *yra viduje*.

Atskiras
: Grąžina 1 (true), jei geometrijos nesidalina jokia erdvės dalimi (nėra persidengimo, nėra susilietimo). Grąžinamas tik 4 apskritimas.

Lygūs
: Grąžina 1 (true), tada ir tik tada, jei geometrijos yra lygiai tokios pačios. Negrąžinami jokie apskritimai.

Liečia
: Tikrina, ar geometrija liečia kitą geometriją. Grąžina 1 (true), jei geometrijos turi bent vieną bendrą tašką, bet jų vidus nepersidengia. Grąžinamas tik 3 apskritimas.

Persidengia
: Tikrina, ar viena geometrija persidengia su kita. Grąžina 1 (true), jei geometrijos dalinasi erdve, turi tokius pačius matmenis, bet nei viena iš jų nėra pilnai kitoje. Grąžinamas tik 2 apskritimas.

Yra viduje

Tikrina, ar geometrija yra kitos viduje. Grąžina 1 (true), jei geometrija a yra pilnai geometrijos b viduje. Grąžinamas tik apskritimas 1.

Kerta

Grąžina 1 (true), jei pateiktos geometrijos turi kelis, bet ne visus, bendrus vidaus taškus ir jų susikirtimas turi mažiau matmenų nei daugiausiai matmenų turinti geometrija. Pavyzdžiui, linijos ir poligono susikirtimas bus linija (true). Poligono ir poligono susikirtimas yra poligonas (false). Iliustracijoje nebūtų grąžinamas nei vienas apskritimas.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Join to features in	`INPUT`	[vektorius: geometrija]	Input vector layer. The output layer will consist of the features of this layer with attributes from matching features in the second layer.
Where the features	`PREDICATE`	[enumeration] [list] Numatytas: [0]	Type of spatial relation the source feature should have with the target feature so that they could be joined. One or more of: • 0 — susikerta • 1 — įtraukia • 2 — equal • 3 — touch • 4 — overlap • 5 — are within • 6 — cross Jei parinkta daugiau nei viena sąlyga, bent viena iš jų (OR veiksmas) turi būti tenkinama, kad geoobjektas būtų ištrauktas.
By comparing to	`JOIN`	[vektorius: geometrija]	The join layer. Features of this vector layer will **add** their attributes to the source layer attribute table if they satisfy the spatial relationship.
Fields to add (leave empty to use all fields) Pasirinktinis	`JOIN_FIELDS`	[lentelės laukas: bet koks] [sąrašas]	Select the specific fields you want to add from the join layer. By default all the fields are added.
Join type	`METHOD`	[sąrašas]	The type of the final joined layer. One of: • 0 — Create separate feature for each matching feature (one-to-many) • 1 — Take attributes of the first matching feature only (one-to-one) • 2 — Take attributes of the feature with largest overlap only (one-to-one)
Discard records which could not be joined	`DISCARD_NONMAT`	[boolean] Numatytas: ne	Remove from the output the input layer's features which could not be joined
Joined field prefix Pasirinktinis	`PREFIX`	[tekstas]	Add a prefix to joined fields in order to easily identify them and avoid field name collision

continues on next page

Table 24.572 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Joined layer Pasirinktinis	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer for the join. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Unjoinable features from first layer Pasirinktinis	`NON_MATCHING`	[toks pat, kaip įvestis] Numatytas: `[Praleisti išvestį]`	Specify the output vector layer for unjoinable features from first layer. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Number of joined features from input table	`JOINED_COUNT`	[skaičius: sveikas skaičius]	
Unjoinable features from first layer Pasirinktinis	`NON_MATCHING`	[toks pat, kaip įvestis]	Vector layer of the non-matched features
Joined layer	`OUTPUT`	[toks pat, kaip įvestis]	Output vector layer with added attributes from the join

Pythono kodas

Algorithm ID: `native:joinattributesbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Join attributes by location (summary)

Takes an input vector layer and creates a new vector layer that is an extended version of the input one, with additional attributes in its attribute table.

The additional attributes and their values are taken from a second vector layer. A spatial criteria is applied to select the values from the second layer that are added to each feature from the first layer.

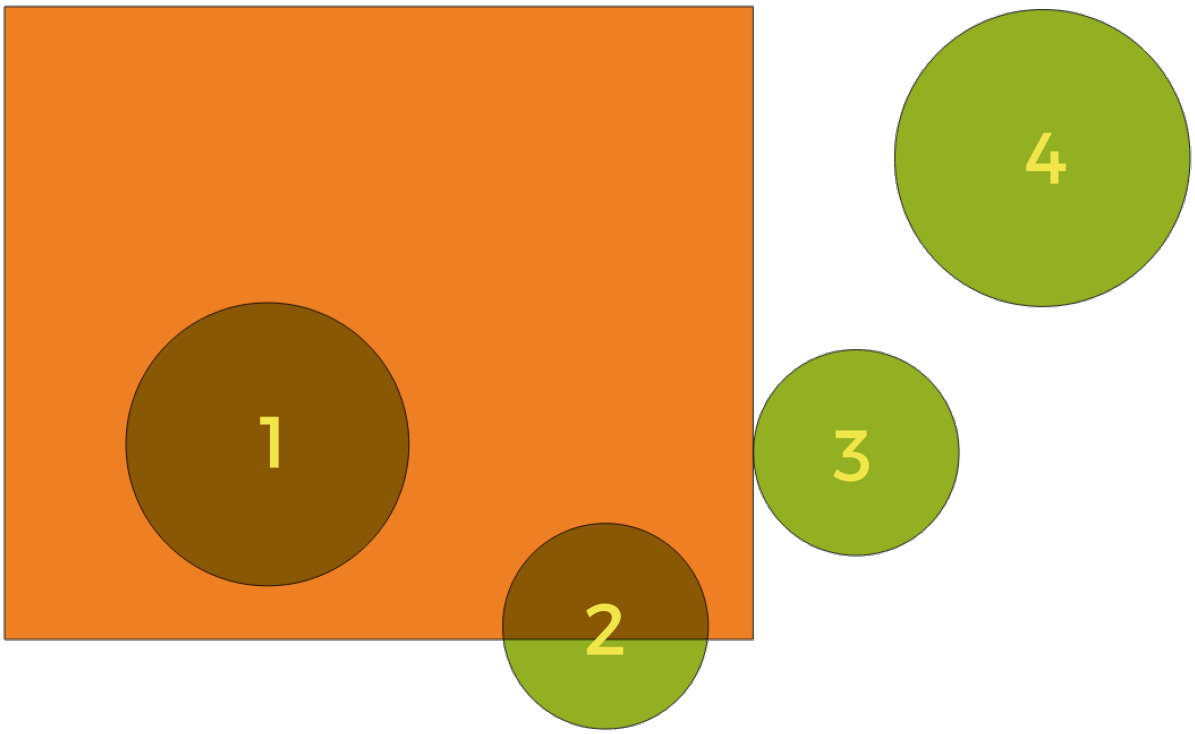
The algorithm calculates a statistical summary for the values from matching features in the second layer (e.g. maximum value, mean value, etc).

Taip pat žiūrėkite

Join attributes by location

Erdvinių ryšių tyrinėjimas

Geometriniai predikatai yra funkcijos, kurios naudojamos nustatant, kokį erdvinį ryšį geoobjektas turi su kitu, lyginant ar ir kaip jų geometrijos dalinasi erdvės dalimi.



24.88 Fig. : Erdvinių ryšių tarp sluoksnių paieška

Naudojant aukščiau pateiktą iliustraciją, mes ieškome žalių apskritimų, erdviškai lygindami juos su oranžiniu stačiakampiu geoobjektus. Galimi geometriniai predikatai yra:

Susikerta
: Tikrina, ar viena geometrija kerta kitą. Grąžina 1 (true), jei geometrijos erdviškai kertasi (dalinasi kažkokia erdvės dalimi - persidengia arba liečia) ir 0, jei to nėra. Iliustracijoje aukščiau True bus grąžinama apskritimams 1, 2 ir 3.

Įtraukia
: Grąžina 1 (true) tada ir tik tada, jei nei vienas b taškas nėra už a ribų, ir bent vienas b vidaus taškas yra a viduje.

Iliustracijoje nei vienas apskritimas nebūtų grąžintas, bet stačiakampis būtų, jei jūs ieškotumėte atvirkščiai, nes pilnai stačiakampyje yra 1 apskritimas. Tai yra priešingybė *yra viduje*.

Atskiras
Grąžina 1 (true), jei geometrijos nesidalina jokia erdvės dalimi (nėra persidengimo, nėra susilietimo). Grąžinamas tik 4 apskritimas.

Lygūs
Grąžina 1 (true), tada ir tik tada, jei geometrijos yra lygiai tokios pačios. Negrąžinami jokie apskritimai.

Liečia
Tikrina, ar geometrija liečia kitą geometriją. Grąžina 1 (true), jei geometrijos turi bent vieną bendrą tašką, bet jų vidus nepersidengia. Grąžinamas tik 3 apskritimas.

Persidengia
Tikrina, ar viena geometrija persidengia su kita. Grąžina 1 (true), jei geometrijos dalinasi erdve, turi tokius pačius matmenis, bet nei viena iš jų nėra pilnai kitoje. Grąžinamas tik 2 apskritimas.

Yra viduje
Tikrina, ar geometrija yra kitos viduje. Grąžina 1 (true), jei geometrija a yra pilnai geometrijos b viduje. Grąžinamas tik apskritimas 1.

Kerta
Grąžina 1 (true), jei pateiktos geometrijos turi kelis, bet ne visus, bendrus vidaus taškus ir jų susikirtimas turi mažiau matmenų nei daugiausiai matmenų turinti geometrija. Pavyzdžiui, linijos ir poligono susikirtimas bus linija (true). Poligono ir poligono susikirtimas yra poligonas (false). Iliustracijoje nebūtų grąžinamas nei vienas apskritimas.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Join to features in	`INPUT`	[vektorius: geometrija]	Input vector layer. The output layer will consist of the features of this layer with attributes from matching features in the second layer.
Where the features	`PREDICATE`	[enumeration] [list] Numatytas: [0]	Type of spatial relation the source feature should have with the target feature so that they could be joined. One or more of: • 0 — susikerta • 1 — įtraukia • 2 — equal • 3 — touch • 4 — overlap • 5 — are within • 6 — cross Jei parinkta daugiau nei viena sąlyga, bent viena iš jų (OR veiksmas) turi būti tenkinama, kad geoobjektas būtų ištrauktas.
By comparing to	`JOIN`	[vektorius: geometrija]	The join layer. Features of this vector layer will **add** summaries of their attributes to the source layer attribute table if they satisfy the spatial relationship.
Fields to summarize (leave empty to use all fields) Pasirinktinis	`JOIN_FIELDS`	[lentelės laukas: bet koks] [sąrašas]	Select the specific fields you want to add from the join layer. By default all the fields are added.

continues on next page

Table 24.574 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Summaries to calculate (leave empty to use all fields) Pasirinktinis	`SUMMARIES`	[enumeration] [list] Default: []	For each input feature, statistics are calculated on joined fields of their matching features. One or more of: • 0 — count • 1 — unique • 2 — min • 3 — max • 4 — range • 5 — sum • 6 — mean • 7 — median • 8 — stddev • 9 — minority • 10 — majority • 11 — q1 • 12 — q3 • 13 — iqr • 14 — empty • 15 — filled • 16 — min_length • 17 — max_length • 18 — mean_length
Discard records which could not be joined	`DISCARD_NONMAT`	[boolean] Numatytas: ne	Remove from the output the input layer's features which could not be joined
Joined layer	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer for the join. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Joined layer	`OUTPUT`	[toks pat, kaip įvestis]	Output vector layer with summarized attributes from the join

Pythono kodas

Algorithm ID: `native:joinbylocationsummary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Join attributes by nearest

Takes an input vector layer and creates a new vector layer with additional fields in its attribute table. The additional attributes and their values are taken from a second vector layer. Features are joined by finding the closest features from each layer.

By default only the nearest feature is joined, but the join can also join to the k-nearest neighboring features.

If a maximum distance is specified, only features which are closer than this distance will be matched.

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Nearest neighbour analysis, *Join attributes by field value*, *Join attributes by location*, *Distance matrix*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties sluoksnis.
Input layer 2	`INPUT_2`	[vektorius: geometrija]	The join layer.
Layer 2 fields to copy (leave empty to copy all fields)	`FIELDS_TO_COPY`	[fields]	Join layer fields to copy (if empty, all fields will be copied).
Discard records which could not be joined	`DISCARD_NONMAT`	[boolean] Numatytas: ne	Remove from the output the input layer records which could not be joined
Joined field prefix	`PREFIX`	[tekstas]	Joined field prefix
Maximum nearest neighbors	`NEIGHBORS`	[skaičius: sveikas skaičius] Numatytas: 1	Maximum number of nearest neighbors
Maximum distance	`MAX_DISTANCE`	[skaičius: slankaus kablelio]	Maximum search distance

continues on next page

Table 24.576 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Joined layer Pasirinktinis	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the vector layer containing the joined features. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Unjoinable features from first layer	`NON_MATCHING`	[toks pat, kaip įvestis] Numatytas: `[Praleisti išvestį]`	Specify the vector layer containing the features that could not be joined. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Joined layer	`OUTPUT`	[toks pat, kaip įvestis]	The output joined layer.
Unjoinable features from first layer	`NON_MATCHING`	[toks pat, kaip įvestis]	Layer containing the features from first layer that could not be joined to any features in the join layer.
Number of joined features from input table	`JOINED_COUNT`	[skaičius: sveikas skaičius]	Number of features from the input table that have been joined.
Number of unjoinable features from input table	`UNJOINABLE_COU`	[skaičius: sveikas skaičius]	Number of features from the input table that could not be joined.

Pythono kodas

Algorithm ID: `native:joinbynearest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

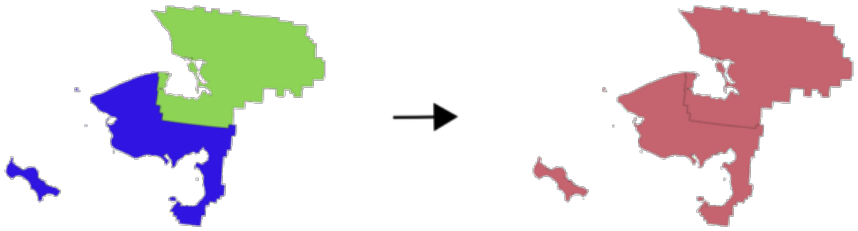
Merge vector layers

Combines multiple vector layers of the **same geometry** type into a single one.

The attribute table of the resulting layer will contain the fields from all input layers. If fields with the same name but different types are found then the exported field will be automatically converted into a string type field. Optionally, new fields storing the original layer name and source can be added.

If any input layers contain Z or M values, then the output layer will also contain these values. Similarly, if any of the input layers are multi-part, the output layer will also be a multi-part layer.

Optionally, the destination coordinate reference system (CRS) for the merged layer can be set. If it is not set, the CRS will be taken from the first input layer. All layers will be reprojected to match this CRS.



Default menu: *Vector ► Data Management Tools*

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Split vector layer

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input Layers	`LAYERS`	[vektorius: bet koks] [sąrašas]	The layers that are to be merged into a single layer. Layers should be of the same geometry type.
Destination CRS Pasirinktinis	`CRS`	[crs]	Choose the CRS for the output layer. If not specified, the CRS of the first input layer is used.
Add source layer information (layer name and path)	`ADD_SOURCE_FIE`	[boolean] Numatytas: Taip	Add fields storing the original layer name and path
Sulietas	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sulietas	`OUTPUT`	[toks pat, kaip įvestis]	Output vector layer containing all the features and attributes from the input layers.

Pythono kodas

Algorithm ID: `native:mergevectorlayers`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Order by expression

Sorts a vector layer according to an expression: changes the feature index according to an expression.

Be careful, it might not work as expected with some providers, the order might not be kept every time.

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Input vector layer to sort
Išraiška	`EXPRESSION`	[išraiška]	Expression to use for the sorting
Rikiuoti didėjančiai	`ASCENDING`	[boolean] Numatytas: Taip	If checked the vector layer will be sorted from small to large values.
Null reikšmės rikiuotės pradžioje	`NULLS_FIRST`	[boolean] Numatytas: ne	If checked, Null values are placed first
Ordered	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Ordered	`OUTPUT`	[toks pat, kaip įvestis]	Output (sorted) vector layer

Pythono kodas

Algorithm ID: `native:orderbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Repair Shapefile

Repairs a broken ESRI Shapefile dataset by (re)creating the SHX file.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input Shapefile	`INPUT`	[file]	Full path to the ESRI Shapefile dataset with a missing or broken SHX file

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Repaired layer	`OUTPUT`	[vektorius: geometrija]	The input vector layer with the SHX file repaired

Pythono kodas

Algorithm ID: `native:repairshapefile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Reproject layer

Reprojects a vector layer in a different CRS. The reprojected layer will have the same features and attributes of the input layer.

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Taip pat žiūrėkite

Priskirti projekciją, *Define projection*, *Find projection*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Input vector layer to reproject
Paskirties CRS	`TARGET_CRS`	[crs] Numatyta: `EPSG:4326 - WGS 84`	Destination coordinate reference system
Convert curved geometries to straight segments	`CONVERT_CURVED`	[boolean] Numatytas: ne	If checked, curved geometries will be converted to straight segments in the process, avoiding potential distortion issues.
Perprojektuota	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Coordinate Operation Pasirinktinis	`OPERATION`	[tekstas]	Specific operation to use for a particular reprojection task, instead of always forcing use of the current project's transformation settings. Useful when reprojecting a particular layer and control over the exact transformation pipeline is required. Requires proj version >= 6. Read more at *Duomenų transformacija*.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Perprojektuota	`OUTPUT`	[toks pat, kaip įvestis]	Output (reprojected) vector layer

Pythono kodas

Algorithm ID: `native:reprojectlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Save vector features to file

Saves vector features to a specified file dataset.

For dataset formats supporting layers, an optional layer name parameter can be used to specify a custom string. Optional GDAL-defined dataset and layer options can be specified. For more information on this, read the online GDAL documentation on the format.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Vector features	`INPUT`	[vektorius: bet koks]	Input vector layer.
Saved features	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Specify the file to save the features to. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Layer name Pasirinktinis	`LAYER_NAME`	[tekstas]	Name to use for the output layer
GDAL dataset options Pasirinktinis	`DATASOURCE_OPT`	[tekstas]	GDAL dataset creation options of the output format. Separate individual options with semicolons.

continues on next page

Table 24.588 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
GDAL layer options Pasirinktinis	LAYER_OPTIONS	[tekstas]	GDAL layer creation options of the output format. Separate individual options with semicolons.
Action to take on pre-existing file	ACTION_ON_EXIS	[sąrašas] Numatytas: 0	How to manage existing features. Valid methods are: • 0 — Create or overwrite file • 1 — Create or overwrite layer • 2 — Append features to existing layer, but do not create new fields • 3 — Append features to existing layer, and create new fields if needed

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Saved features	OUTPUT	[toks pat, kaip įvestis]	Vector layer with the saved features.
File name and path	FILE_PATH	[tekstas]	Output file name and path.
Layer name	LAYER_NAME	[tekstas]	Name of the layer, if any.

Pythono kodas

Algorithm ID: `native:savefeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Set layer encoding

Sets the encoding used for reading a layer's attributes. No permanent changes are made to the layer, rather it affects only how the layer is read during the current session.

Pastaba

Changing the encoding is only supported for some vector layer data sources.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Saved features	`INPUT`	[vektorius: geometrija]	Vector layer to set the encoding.
Koduotė	`ENCODING`	[tekstas]	Text encoding to assign to the layer in the current QGIS session.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties sluoksnis	`OUTPUT`	[toks pat, kaip įvestis]	Input vector layer with the set encoding.

Pythono kodas

Algorithm ID: `native:setlayerencoding`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Split features by character

Features are split into multiple output features by splitting a field's value at a specified character. For instance, if a layer contains features with multiple comma separated values contained in a single field, this algorithm can be used to split these values up across multiple output features. Geometries and other attributes remain unchanged in the output. Optionally, the separator string can be a regular expression for added flexibility.

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties vektorinis sluoksnis
Split using values in the field	`FIELD`	[tablefield: any]	Field to use for splitting
Split value using character	`CHAR`	[tekstas]	Character to use for splitting

continues on next page

Table 24.592 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Use regular expression separator	`REGEX`	[boolean] Numatytas: ne	
Split	`OUTPUT`	[toks pat, kaip įvestis] Default: `Create temporary layer`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Split	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer.

Pythono kodas

Algorithm ID: `native:splitfeaturesbycharacter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Split vector layer

Creates a set of vectors in an output folder based on an input layer and an attribute. The output folder will contain as many layers as the unique values found in the desired field.

The number of files generated is equal to the number of different values found for the specified attribute.

It is the opposite operation of *merging*.

Default menu: *Vector ► Data Management Tools*

Taip pat žiūrėkite

Merge vector layers

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties vektorinis sluoksnis
Unique ID field	`FIELD`	[tablefield: any]	Field to use for splitting
Išvesties aplankas	`OUTPUT`	[aplankas] Numatytas: `[Įrašyti į laikiną aplanką]`	Specify the directory for the output layers. *One of*: • Įrašyti į laikiną aplanką • Įrašyti į aplanką

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Output file type Pasirinktinis	`FILE_TYPE`	[sąrašas] Default: `gpkg` in the dialog window	Select the extension of the output files. If not specified or invalid, the output files format will be the one set in the „Default output vector layer extension“ Processing setting.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties aplankas	`OUTPUT`	[aplankas]	The directory for the output layers
Output layers	`OUTPUT_LAYERS`	[same as input] [list]	The output vector layers resulting from the split.

Pythono kodas

Algorithm ID: `native:splitvectorlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Truncate table

Truncates a layer, by deleting all features from within the layer.

> **Įspėjimas**
>
> This algorithm modifies the layer in place, and deleted features cannot be restored!

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties vektorinis sluoksnis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Truncated layer	`OUTPUT`	[aplankas]	The input layer, all features deleted

Pythono kodas

Algorithm ID: `native:truncatetable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.26 Vektorinė geometrija

Pridėti geometrijos atributus

Paskaičiuoja vektorinio sluoksnio geoobjektų geometrijos savybes ir įtraukia jas į išvesties sluoksnį.

Sukuria naują sluoksnį su tokiu pačiu turiniu, kaip ir įvesties, bet su papildomais atributais, kuriuose yra geometriniai matavimai paremti nurodyta CRS.

Į lentelę pridedami atributai priklauso nuo įvesties sluoksnio geometrijos tipo ir matmenų:

- **taškų** sluoksniams: X (`xcoord`), Y (`ycoord`), Z (`zcoord`) koordinatės ir/ar M reikšmė (`mvalue`)
- **linijos** sluoksniams: `length`, o LineString ir CompoundCurve geometrijos tipams dar ir `sinuosity` bei tiesus atstumas (`straightdis`)
- **poligonų** sluoksniams: `perimeter` ir `area`

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.

Numatytas meniu: *Vektorius ► Geometrijos įrankiai*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Skaičiuoti naudojant	`CALC_METHOD`	[sąrašas] Numatytas: 0	Geometrinių savybių skaičiavimo parametrai. Vienas iš: • 0 — Sluoksnio CRS • 1 — Projekto CRS • 2 — Elipsoidinis
Pridėta geometrinė informacija	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties (įvesties kopijos su geometrija) sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Pridėta geometrinė informacija	`OUTPUT`	[toks pat, kaip įvestis]	Įvesties vektorinio sluoksnio kopija su geometrijos laukų papildymais

Pythono kodas

Algorithm ID: `native:exportaddgeometrycolumns`

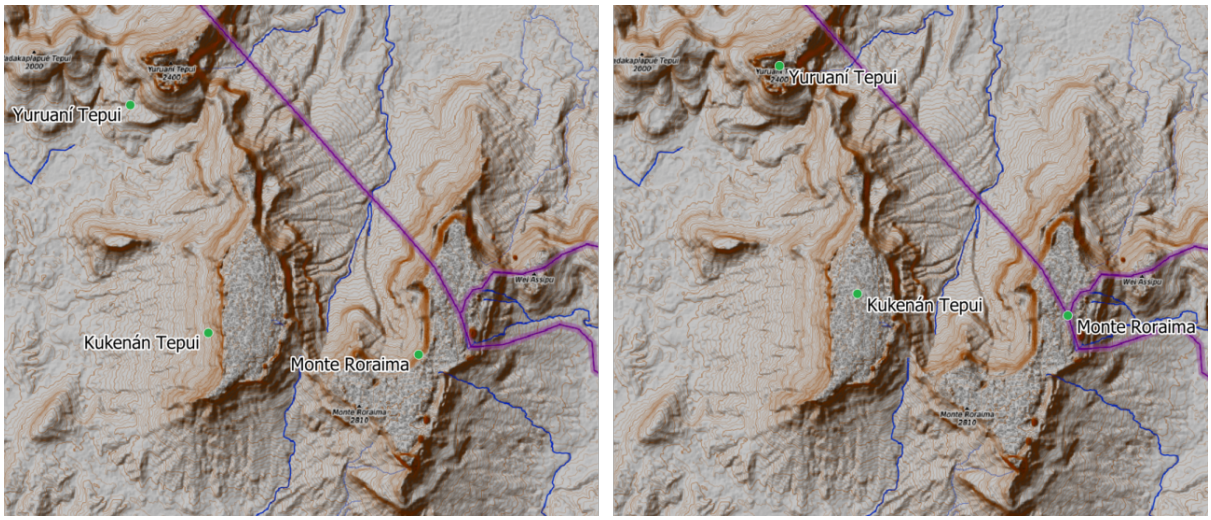
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Paprastoji transformacija

Taiko sluoksnio geometrijoms paprastą transformaciją. Paprastos transformacijos atlieka vertimą, mastelio keitimą ir pasukimą. Veiksmai vykdomi šia seka: mastelio keitimas, pasukimas ir vertimas.

Z ir M reikšmės (jei jos yra) gali būtų verčiamos ir keičiamas jų mastelis.



24.89 Fig. : Vektorinis taškų sluoksnis (žali taškai) prieš (kairėje) ir po (dešinėje) paprastos transformacijos (vertimo).

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Taip pat žiūrėkite

Translate

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Vertimas (x-ašis)	`DELTA_X`	[skaičius: slankaus kablelio] Numatytas: 0.0	X ašyje taikomas poslinkis.
Vertimas (y-ašis)	`DELTA_Y`	[skaičius: slankaus kablelio] Numatytas: 0.0	Y ašyje taikomas poslinkis.
Vertimas (z-ašis)	`DELTA_Z`	[skaičius: slankaus kablelio] Numatytas: 0.0	Z ašyje taikomas poslinkis.
Vertimas (m-reikšmės)	`DELTA_M`	[skaičius: slankaus kablelio] Numatytas: 0.0	M reikšmėms taikomas poslinkis.

continues on next page

Table 24.601 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Mastelio koeficientas (x-ašis)	`SCALE_X`	[skaičius: slankaus kablelio] Numatytas: 1.0	Mastelio keitimo reikšmė (išplėtimas ar sutraukimas), kuri taikoma X ašyje.
Mastelio koeficientas (y-ašis)	`SCALE_Y`	[skaičius: slankaus kablelio] Numatytas: 1.0	Mastelio keitimo reikšmė (išplėtimas ar sutraukimas), kuri taikoma Y ašyje.
Mastelio koeficientas (z-ašis)	`SCALE_Z`	[skaičius: slankaus kablelio] Numatytas: 1.0	Mastelio keitimo reikšmė (išplėtimas ar sutraukimas), kuri taikoma Z ašyje.
Mastelio koeficientas (m-reikšmės)	`SCALE_M`	[skaičius: slankaus kablelio] Numatytas: 1.0	Mastelio keitimo reikšmė (išplėtimas ar sutraukimas), kuri taikoma m reikšmėms.
Sukimas aplink z-ašį (laipsniai prieš laikrodžio rodyklę)	`ROTATION_Z`	[skaičius: slankaus kablelio] Numatytas: 0.0	Pasukimo kampas laipsniais.
Transformuotas	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Transformuotas	`OUTPUT`	[toks pat, kaip įvestis]	Išvesties (transformuotas) vektorinis sluoksnis.

Pythono kodas

Algoritmo ID: `native:affinetransform`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Agreguoti

Ima vektorių ar lentelės sluoksnį ir sukuria naują sluoksnį agreguojant geoobjektus pagal `group by` išraišką.

Geoobjektai, kuriems `group by` išraiška grąžina tą pačią reikšmę, grupuojami kartu.

Galima grupuoti visus šaltinio geoobjektus naudojant parametrui `group by` konstantą, pavyzdžiui: NULL.

Taipogi galima grupuoti geoobjektus pagal kelis laukus naudojant funkciją Array, pavyzdžiui: Array(„Laukas1“, „Laukas2“).

Geometrijos (jei jos yra) kiekvienai grupei kombinuojamos į vieną kelių dalių geometriją. Išvesties atributai skaičiuojami priklausomai nuo kiekvienos nurodytos agregavimo funkcijos.

Šis algoritmas leidžia naudoti numatytas QGIS išraiškų variklio *agregavimo funkcijas*.

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.

Taip pat žiūrėkite

Collect geometries, *Dissolve*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties vektorinis sluoksnis
Group by išraiška	`GROUP_BY`	[lentelės laukas: bet koks] Numatyta: „NULL“	Parinkite grupavimo lauką. Jei *NULL* - bus grupuojami visi geoobjektai.

continues on next page

Table 24.603 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Agregatai	`AGGREGATES`	[sąrašas]	Sąrašas išvesties sluoksnio laukų apibrėžimų. Lauko apibrėžimo pavyzdys: *{‚aggregate‘: ‚sum‘, ‚delimiter‘: „‘, ‚input‘: , $area‘, ‚length‘: 10, ‚name‘: ‚totarea‘, ‚precision‘: 0, ‚type‘: 6}* Pagal nutylėjimą sąraše yra visi įvesties sluoksnio laukai. GUI leidžia jums keisti šiuos laukus ir jų apibrėžimus, jūs taipogi galite: • Spausti mygtuką ir pridėti naują lauką. • Spausti mygtuką ir ištrinti pažymėtą lauką. • Naudoti ir laukų rikiuotės keitimui. • Spausti , kad atstatytumėte pradines reikšmes (įvesties sluoksnio laukai). Kiekvienam laukui, iš kurio norite ištraukti informaciją, jums reikia nurodyti: ***Įvesties išraišką* (`input`) [išraiška]** Įvesties sluoksnio lauką ar išraišką. ***Agregavimo funkcija* (`aggregate`) [iš sąrašo]** Funkcija, kurią reikia taikyti įvesties išraiškai, kad grąžintų agreguotą reikšmę. Paprastai tai būna *Agregavimo funkcijos*, plius tam skirtos „first_value“ ir „last_value“ funkcijos rikiuotiems elementų sąrašams. Numatyta: *concatenate* (teksto ir datos tipams), *sum* (skaitiniams duomenų tipams) ***Skirtukas* (`delimiter`) [tekstas]** Tekstas, kuriuo atskirti agreguotas reikšmes, pavyzdžiui sudedant visas reikšmes. Numatytasis: , ***Išvesties lauko pavadinimas* (`name`) [tekstas]** Agreguoto lauko pavadinimas išvesties sluoksnyje. Pagal nutylėjimą jis toks pat, kaip ir įvesties sluoksnyje. ***Tipas* (`type`) [iš sąrašo]** Išvesties lauko duomenų tipas. Galimi tipai gali būti nesuderinami su išvesties sluoksnio tiekėju. Vienas iš: **Dėmesio** Kai kuriems tipams, pvz. sąrašams, papildomas parametras `sub_type` leidžia tiksliau apibūdinti konkretų duomenų tipą. Jis automatiškai nustatomas GUI, bet gali būti naudingas, jei jūs paleidžiate algoritmą Python’e

Table 24.603 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Įkelti laukus iš sluoksnio	Tik GUI	[vektorius: bet koks]	Galite įkelti laukus iš kito sluoksnio ir naudoti juos agregavimui
Agreguoti	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties (agreguotą) sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Agreguoti	`OUTPUT`	[toks pat, kaip įvestis]	Daugiageometrinis vektorinis sluoksnis su agreguotomis reikšmėmis

Pythono kodas

Algoritmo ID: `native:aggregate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

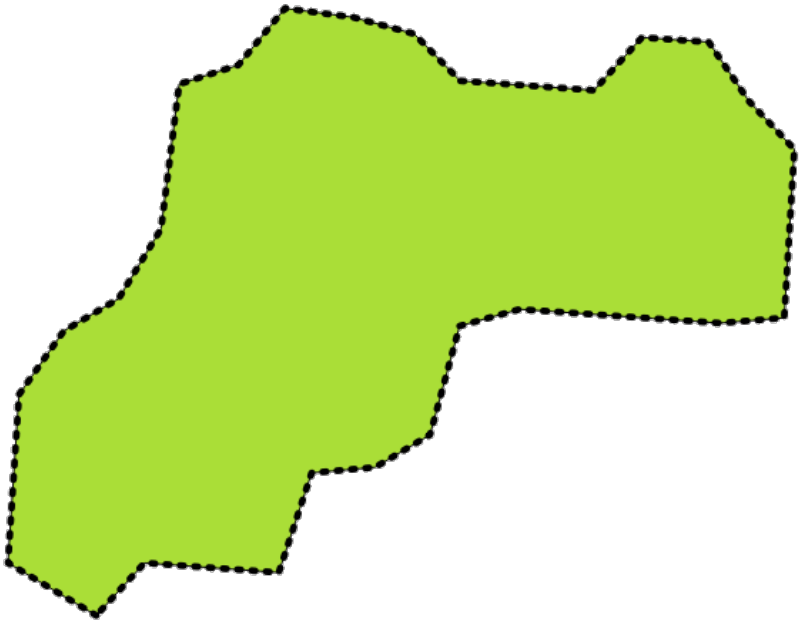
algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Ribos

Grąžina įvesties geometrijų kombinatorinių ribų uždarą lotą (t.y. geometrijos topologinę ribą).

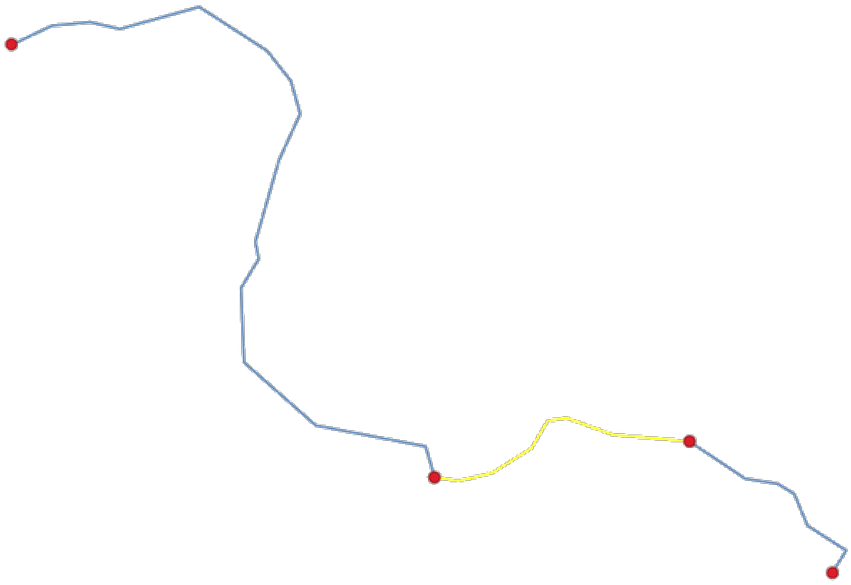
Tik poligonų ir linijų sluoksniams

Poligonų geometrijų riba sudaryta iš visų linijų, sudarančių poligono žiedus.



24.90 Fig. : Pradinio poligonų sluoksnio ribos (juodos punktyrinės linijos)

**Linijų geometrijų“ ribos yra jų pabaigos taškai.



24.91 Fig. : Linijų ribų sluoksnis (raudoni taškai). Geltonas - parinktas geoobjektas.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija, poligonas]	Įvesties linijų ar poligonų vektorinis sluoksnis
Riba	`OUTPUT`	[vektorius: taškai, linijos] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties (ribų) sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Riba	`OUTPUT`	[vektorius: taškai, linijos]	Įvesties ribų sluoksnis (taškai linijoms, linija poligonams)

Pythono kodas

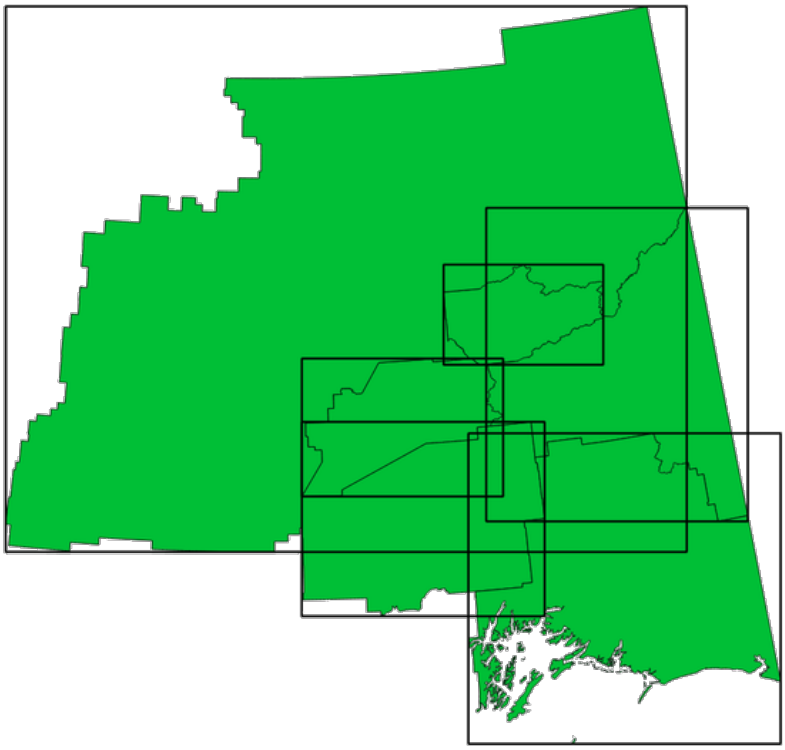
Algoritmo ID: `native:boundary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Apimties stačiakampiai

Skaičiuoja kiekvieno įvesties sluoksnio geoobjekto apimties stačiakampį (voką). Palaikomos poligonų ir linijų geometrijos.



24.92 Fig. : Juodos linijos vaizduoja kiekvieno poligono geoobjekto apimties stačiakampį

Leidžia *vietoje keisti* poligonų geoobjektus

Taip pat žiūrėkite

Minimum bounding geometry

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija, poligonas]	Įvesties linijų ar poligonų vektorinis sluoksnis
Aprėptis	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties (apimties stačiakampių) sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Aprėptis	`OUTPUT`	[vektorius: poligonas]	Įvesties sluoksnio apimties stačiakampiai. Be įvesties atributų išvesties sluoksnyje taipogi yra šie laukai: sukurto poligono `width`, `height`, `area` ir `perimeter`.

Pythono kodas

Algoritmo ID: `native:boundingboxes`

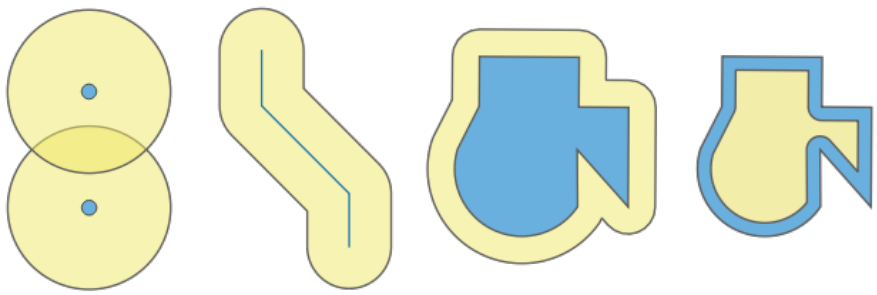
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Buferis

Skaičiuoja buferio plotą aplink visus įvesties sluoksnio geoobjektus, naudojant fiksuotą arba nuo duomenų priklausantį atstumą.

Poligono įvesties sluoksniams galima naudoti neigiamą atstumą. Tokiu atveju buferis reikš mažesnį poligoną (sutrauktą).



24.93 Fig. : Buferiai (geltoni) taškų, linijų, poligonų su teigiamu buferiu ir poligonas su neigiamu buferiu

Leidžia *vietoje keisti* poligonų geoobjektus

Numatytas meniu: *Vektorius* ► *Geometrijos įrankiai*

Įspėjimas

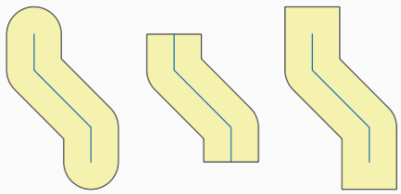
Šis algoritmas gali išmesti esamus pirminius raktus ar FID reikšmes ir perkurti juos išvesties sluoksniuose, priklausomai nuo to, kokie įvesties parametrai.

Taip pat žiūrėkite

Variable distance buffer, *Multi-ring buffer (constant distance)*, *Variable width buffer (by M value)*

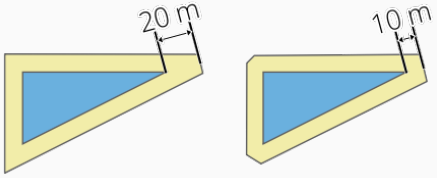
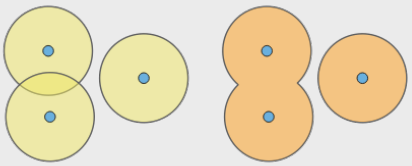
Parametrai

Baziniai parametrai

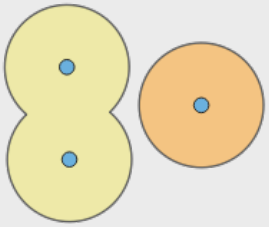
Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Atstumas	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 10.0	Buferio atstumas (nuo kiekvieno geoobjekto ribos). Galite naudoti dešinėje pusėje esantį mygtuką, jei norite nustatyti nuo lauko duomenų priklausantį spindulį skaičiavimui. Tokiu būdu jūs galite skirtingiems geoobjektams turėti skirtingą spindulį.
Segmentai	`SEGMENTS`	[skaičius: sveikas skaičius] Numatytas: 5	Valdo skaičių linijų segmentų, kurie naudojami aproksimuojant apskritimo ketvirčius kuriant užapvalintus poslinkius.
Pabaigos stilius	`END_CAP_STYLE`	[sąrašas] Numatytas: 0	Nurodo kaip valdomos buferio linijų pabaigos. Vienas iš: • 0 — Apvalūs • 1 — Plokščias • 2 — Kvadratinis  24.94 Fig. : Apvali, plokščia arba kvadratinė pabaiga
Jungimo stilius	`JOIN_STYLE`	[sąrašas] Numatytas: 0	Nurodo, ar paslinkus linijos kampus reikia naudoti apvalias, mitros ar kampines jungtis. Parinktys yra: • 0 — Apvalūs • 1 — Mitra • 2 — Kampas 24.95 Fig. : Apvalūs, mitros ir kampo jungimo stiliai

continues on next page

Table 24.609 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Mitros riba	`MITER_LIMIT`	[skaičius: slankaus kablelio] Numatyta: 2.0	Nustato maksimalų atstumą nuo paslenkamos geometrijos kuriant mitros jungtį. Nurodomas kaip poslinkio atstumo koeficientas (taikoma tik mitros jungimo stiliui). Minimalus: 1.0  24.96 Fig. : 10m buferis su 2 riba ir 10m buferis su 1 riba
Sulieti rezultatą	`DISSOLVE`	[loginis] Numatytas: ne	Sulieti gautą buferį. Jei `True` (įjungta), persidengiantys buferiai bus sulieti (sujungti) į vieną kelių dalių geoobjektą.  24.97 Fig. : Standartinis (trys vienos dalies geoobjektai - kairėje), sulietas (1 kelių dalių geoobjektas su 2 dalimis - dešinėje)
Buferiai	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties (buferių) sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Palikti nesusijusius geoobjektus atskirais	`SEPARATE_DISJO`	[loginis] Numatytas: ne	Jei `True` (įjungta) ir įjungtas suliejimas, geoobjektai, kurie nepersidengia ir nesiliečia, bus eksportuojami kaip atskiri geoobjektai (o ne kelių dalių geoobjekto dalys).  24.98 Fig. : Rezultatai kaip 2 vienos dalies geoobjektai

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Buferiai	`OUTPUT`	[vektorius: poligonas]	Išvesties (buferių) poligonų sluoksnis

Pythono kodas

Algoritmo ID: `native:buffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

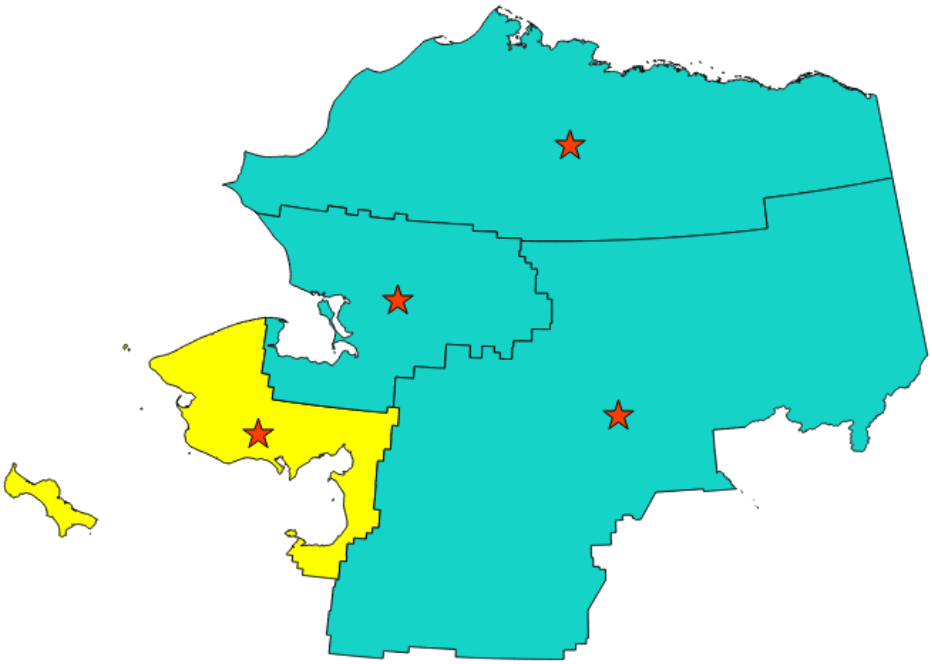
algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Centroidai

Sukuria naują taškų sluoksnį su taškais, reprezentuojančiais įvesties sluoksnio geometrijos centroidus.

Centroidas yra vienas taškas, reprezentuojantis geoobjekto (visų dalių) svorio centrą, taigi jis gali būti už geoobjekto ribų. Bet taipogi gali būti ir taškas ant bet kurios geoobjekto dalies.

Išvesties sluoksnio taškų atributai bus tokie patys, kaip pradinių geoobjektų.



24.99 Fig. : Raudonos žvaigždės reprezentuoja įvesties sluoksnio geoobjektų centroidus.

Leidžia taškų *geoobjektus keisti vietoje*

Numatytas meniu: *Vektorius* ► *Geometrijos įrankiai*

Įspėjimas

Šis algoritmas gali išmesti esamus pirminius raktus ar FID reikšmes ir perkurti juos išvesties sluoksniuose, priklausomai nuo to, kokie įvesties parametrai.

Taip pat žiūrėkite

Point on Surface

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Kurti centroidą kiekvienai daliai	`ALL_PARTS`	[loginis] Numatytas: ne	Jei True (įjungtas), centroidas bus kuriamas kiekvienai geometrijos daliai

continues on next page

Table 24.612 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Centroidai	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties (centroidų) sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Centroidai	`OUTPUT`	[vektorius: taškas]	Išvesties vektorinis taškų sluoksnis (centroidai)

Pythono kodas

Algoritmo ID: `native:centroids`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Tikrinti tvarkingumą

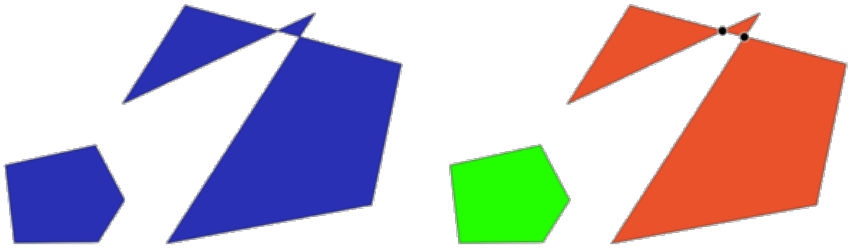
Vykdo vektorinio sluoksnio geometrijų tvarkingumo tikrinimą.

Geometrijos klasifikuojamos į tris grupes (tvarkingos, netvarkingos ir klaidos) ir kiekvienai grupei sukuriamas vektorinis sluoksnis su geoobjektais:

- **Tvarkingos išvesties** sluoksnyje yra tik tvarkingi geoobjektai (be topologinių klaidų).
- **Netvarkingos išvesties** sluoksnyje yra visi algoritmo aptikti netvarkingi geoobjektai.
- **Klaidų išvesties** sluoksnyje yra taškai, kur buvo rasti netvarkingi geoobjektai.

Sukurtų sluoksnių atributų lentelėse bus pridėta šiek tiek papildomos informacijos („message“ prie **klaidų** sluoksnio, „FID“ ir „_errors“ prie **netvarkingų** sluoksnio ir tik „FID“ prie **tvarkingų** sluoksnio):

Kiekvieno sukurto vektorinio sluoksnio atributų lentelėje bus šiek tiek papildomos informacijos (rastų klaidų skaičius ir jų tipai):



24.100 Fig. : Kairėje: įvesties sluoksnis. Dešinėje: tinkamų sluoksnis (žalias), netinkamų sluoksnis (oranžinis)

Numatytas meniu: *Vektorius* ► *Geometrijos įrankiai*

Taip pat žiūrėkite

Check Geometry, *Fix geometries*, *Validate coverage*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT_LAYER`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Metodas	`METHOD`	[sąrašas] Numatyta: 2	Tinkamumo tikrinimo metodas. Parinktys: • 0: Parinktas skaitmeninimo nustatymuose • 1: QGIS • 2: GEOS
Ignoruoti žiedų susikirtimą	`IGNORE_RING_SE`	[loginis] Numatytas: ne	Ignoruoti save kertančius žiedus tikrinant tinkamumą.
Tvarkingų išvestis	`VALID_OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite vektorinį sluoksnį, kuriame bus pradinio sluoksnio tvarkingų geoobjektų kopijos. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Netvarkingų išvestis	`INVALID_OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Vektorinis sluoksnis, kuriame bus pradinio sluoksnio netvarkingų geoobjektų kopijos su lauke `_errors` išvardinta rastos klaidos(ų) santrauka. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

continues on next page

Table 24.614 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Klaidų išvestis	`ERROR_OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Taškų sluoksnis su tiksliomis aptiktų tvarkingumo problemų pozicijomis, lauke `message` aprašant rastą klaidą(as). *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Klaidų skaičius	`ERROR_COUNT`	[skaičius: sveikas skaičius]	Klaidas sukėlusių geometrijų skaičius.
Klaidų išvestis	`ERROR_OUTPUT`	[vektorius: taškas]	Taškų sluoksnis su tiksliomis aptiktų tinkamumo problemų pozicijomis ir lauke `message` aprašyta rasta klaida(omis).
Netvarkingų geoobjektų skaičius	`INVALID_COUNT`	[skaičius: sveikas skaičius]	Netvarkingų geometrijų skaičius.
Netvarkingų išvestis	`INVALID_OUTPUT`	[toks pat, kaip įvestis]	Vektorinis sluoksnis su pradinio sluoksnio netvarkingų geoobjektų kopijomis ir lauke `_errors` išvardinta rastos klaidos(ų) santrauka.
Tvarkingų geoobjektų skaičius	`VALID_COUNT`	[skaičius: sveikas skaičius]	Tvarkingų geometrijų skaičius.
Tvarkingų išvestis	`VALID_OUTPUT`	[toks pat, kaip įvestis]	Vektorinis sluoksnis su pradinio sluoksnio tvarkingų geoobjektų kopijomis.

Pythono kodas

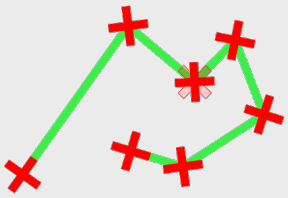
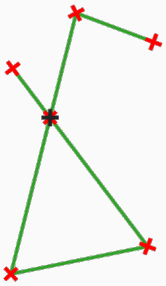
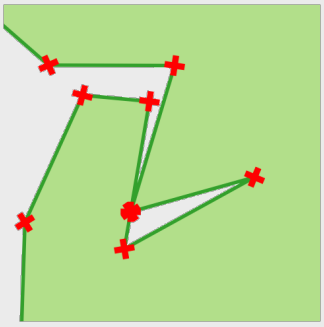
Algorithm ID: `native:checkvalidity`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

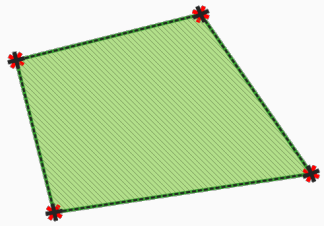
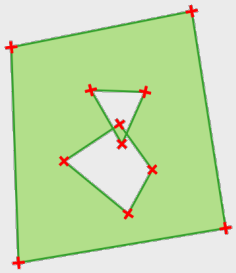
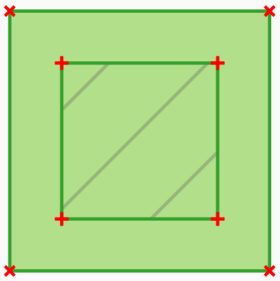
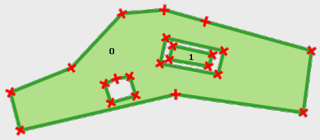
Klaidų pranešimų tipai ir jų reikšmės

24.616 Table : Keli QGIS geometrijos tikrinimo klaidų pavyzdžiai

Geometrijos kontekstas	GEOS tikrinimas ir klaidos pranešimas	QGIS tikrinimas ir klaidos pranešimas
Vienas po kito einantys linijos taškai turi tas pačias koordinates 	*Pakartotas taškas*	*Linijoje yra x pasikartojančių taškų ties b*
Linijos segmentai kerta vienas kitą 		*Linijos c segmentai a ir b kertasi ties d*
Poligono geometrija liečia save ir sukuria žiedą 	*Žiedo savęs kirtimas*	*Žiedo savęs kirtimas*

continues on next page

Table 24.616 – tęsinys iš praeito puslapio

Geometrijos kontekstas	GEOS tikrinimas ir klaidos pranešimas	QGIS tikrinimas ir klaidos pranešimas
Du poligono geometrijos žiedai (išorinis ir vidinis) yra identiški 	*Pasikartojantys žiedai*	
Geometrija liečia pati save 	*Savęs kirtimas*	
Poligono geometrija yra virš kitos poligono geometrijos 	*Vidinis kiautas*	
Dalis MultiPoligono geometrijos yra MultiPoligono geometrijos skylėje 		*Poligonas a yra poligono b viduje*

continues on next page

Table 24.616 – tęsinys iš praeito puslapio

Geometrijos kontekstas	GEOS tikrinimas ir klaidos pranešimas	QGIS tikrinimas ir klaidos pranešimas
Taško geometrija neturi tinkamos koordinačių poros. Koordinačių poroje nėra platumos reikšmės ir ilgumos reikšmės tokioje rikiuotėje.	*Netinkama koordinatė*	

Collect geometries

Takes a vector layer and collects its geometries into new multipart geometries.

One or more attributes can be specified to collect only geometries belonging to the same class (having the same value for the specified attributes), alternatively all geometries can be collected.

All output geometries will be converted to multi geometries, even those with just a single part. This algorithm does not dissolve overlapping geometries - they will be collected together without modifying the shape of each geometry part.

See the ‚Promote to multipart' or ‚Aggregate' algorithms for alternative options.

Numatytas meniu: *Vektorius* ► *Geometrijos įrankiai*

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Agreguoti, *Promote to multipart*, *Dissolve*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Unique ID fields	`FIELD`	[lentelės laukas: bet koks] [sąrašas]	Choose one or more attributes to collect the geometries
Collected	`OUTPUT`	[toks pat, kaip įvestis]	Vector layer with collected geometries

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Collected	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer for the collected geometries. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Pythono kodas

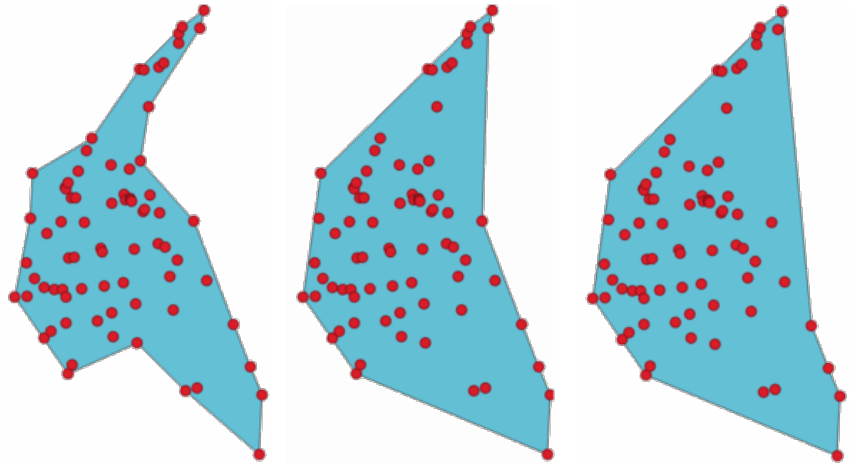
Algorithm ID: `native:collect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Concave hull

Computes the concave hull of the features from an input point layer.



24.101 Fig. : Concave hulls with different thresholds (0.3, 0.6, 0.9)

Taip pat žiūrėkite

Convex hull

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input point layer	`INPUT`	[vektorius: taškas]	Input point vector layer
Threshold	`ALPHA`	[skaičius: slankaus kablelio] Default: 0.3	Number from 0 (maximum concave hull) to 1 (convex hull).
Allow holes	`HOLES`	[loginis] Numatytas: Taip	Choose whether to allow holes in the final concave hull
Split multipart geometry into singlepart geometries	`NO_MULTIGEOMET`	[loginis] Numatytas: ne	Check if you want to have singlepart geometries instead of multipart ones.
Concave hull	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Concave hull	`OUTPUT`	[vektorius: poligonas]	Išvesties vektorinis sluoksnis

Pythono kodas

Algorithm ID: `native:concavehull`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Convert geometry type

Generates a new layer based on an existing one, with a different type of geometry.

The attribute table of the output layer is the same as the one of the input layer.

Not all conversions are possible. For instance, a line layer can be converted to a point layer, but a point layer cannot be converted to a line layer.

Taip pat žiūrėkite

Polygonize, *Lines to polygons*, *Polygons to lines*, *Points to path*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
New geometry type	`TYPE`	[sąrašas] Numatytas: 0	Geometry type to apply to the output features. One of: • 0 — Centroids • 1 — Nodes • 2 — Linestrings • 3 — Multilinestrings • 4 — Polygons
Pakeistas	`OUTPUT`	[vektorius: geometrija] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Pakeistas	`OUTPUT`	[vektorius: geometrija]	Output vector layer - the type depends on the parameters

Pythono kodas

Algorithm ID: `native:convertgeometrytype`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Convert to curved geometries

Converts a geometry into its curved geometry equivalent.

Already curved geometries will be retained without change.

Allows *features in-place modification* of line and polygon features

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vector: line or polygon]	Įvesties vektorinis sluoksnis
Maximum distance tolerance	`DISTANCE`	[skaičius: slankaus kablelio] Default: 0.000001	The maximum distance allowed between the original location of vertices and where they would fall on the converted curved geometries
Maximum angle tolerance	`ANGLE`	[skaičius: slankaus kablelio] Default: 0.000001	Segments are considered as suitable for replacing with an arc if the points are all regularly spaced on the candidate arc. This parameter specifies the maximum angular deviation (in degrees) allowed when testing for regular point spacing. Between 0 and 45°.
Curves	`OUTPUT`	[vector: compoundcurve or curvepolygon] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Curves	`OUTPUT`	[vector: compoundcurve or curvepolygon]	Output vector layer with curved geometries

Pythono kodas

Algorithm ID: `native:converttocurves`

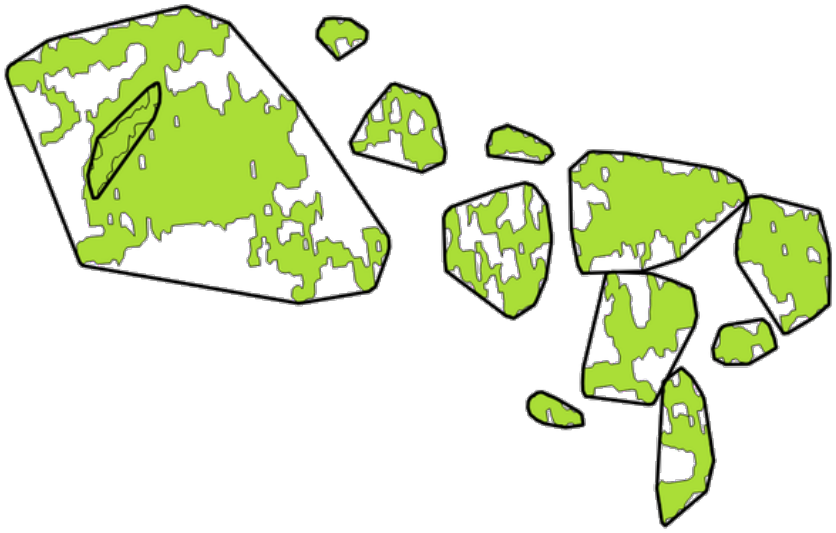
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Convex hull

Calculates the convex hull for each feature in an input layer.

See the ‚Minimum bounding geometry' algorithm for a convex hull calculation which covers the whole layer or grouped subsets of features.



24.102 Fig. : Black lines identify the convex hull for each layer feature

Leidžia *vietoje keisti* poligonų geoobjektus

Numatytas meniu: *Vektorius* ► *Geometrijos įrankiai*

Taip pat žiūrėkite

Minimum bounding geometry, *Concave hull*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Convex hull	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Convex hull	`OUTPUT`	[vektorius: poligonas]	The output (convex hull) vector layer. Other than the input attributes, the output layer also contains the following fields: `area` and `perimeter` of the generated polygon.

Pythono kodas

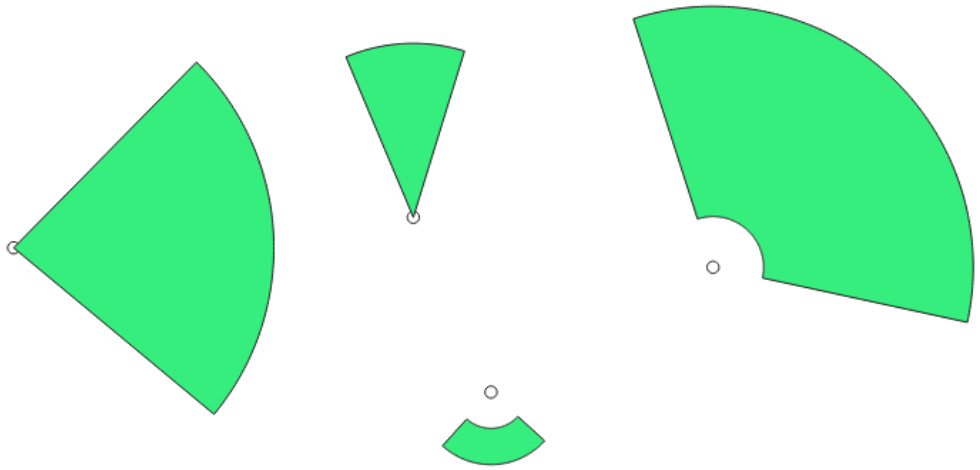
Algorithm ID: `native:convexhull`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Create wedge buffers

Creates wedge shaped buffers from input points.



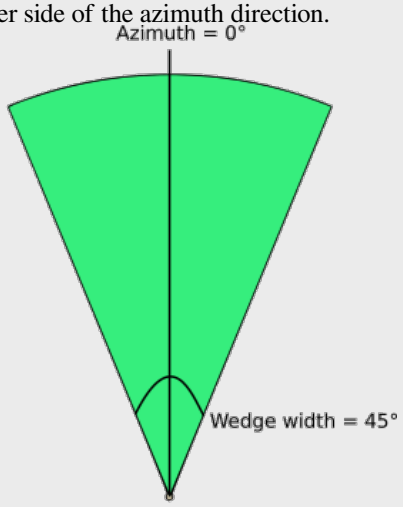
24.103 Fig. : Wedge buffers

The native output from this algorithm are CurvePolygon geometries, but these may be automatically segmentized to Polygons depending on the output format.

Taip pat žiūrėkite

Buferis, *Variable width buffer (by M value)*, *Tapered buffers*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: taškas]	Input point vector layer
Azimuth (degrees from North)	`AZIMUTH`	[skaičius: slankaus kablelio] Numatytas: 0.0	Angle (in degrees) as the middle value of the wedge
Wedge width (in degrees)	`WIDTH`	[skaičius: slankaus kablelio] Default: 45.0	Width (in degrees) of the buffer. The wedge will extend to half of the angular width either side of the azimuth direction.  24.104 Fig. : Azimuth and width values of the wedge buffer
Outer radius	`OUTER_RADIUS`	[skaičius: slankaus kablelio] Numatytas: 1.0	The outer *size* (length) of the wedge: the size is meant from the source point to the edge of the wedge shape.
Inner radius Pasirinktinis	`INNER_RADIUS`	[skaičius: slankaus kablelio] Numatytas: 0.0	Inner radius value. If 0 the wedge will begin from the source point.
Buffers	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Buffers	`OUTPUT`	[vektorius: poligonas]	The output (wedge buffer) vector layer

Pythono kodas

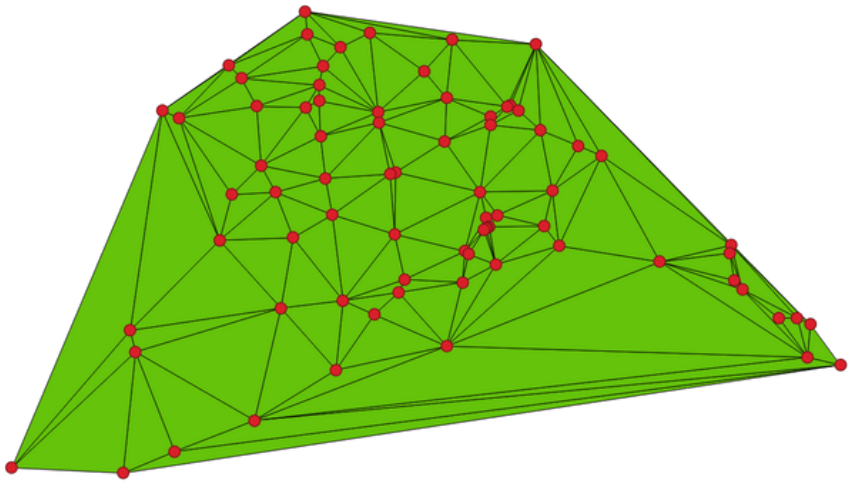
Algorithm ID: `native:wedgebuffers`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Delaunay triangulation

Creates a polygon layer with the Delaunay triangulation corresponding to the input point layer.



24.105 Fig. : Delaunay triangulation on points

Numatytas meniu: *Vektorius* ► *Geometrijos įrankiai*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: taškas]	Input point vector layer
Tolerancija Pasirinktinis	`TOLERANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Specifies an optional snapping tolerance which can be used to improve the robustness of the triangulation.

continues on next page

Table 24.629 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Add point IDs to output	`ADD_ATTRIBUTES`	[loginis] Numatytas: Taip	Specifies whether fields storing involved point features ID should be added to the output. If False, an `id` field is used to identify the polygons.
Delaunay triangulation	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Delaunay triangulation	`OUTPUT`	[vektorius: poligonas]	The output (Delaunay triangulation) vector layer

Pythono kodas

Algorithm ID: `native:delaunaytriangulation`

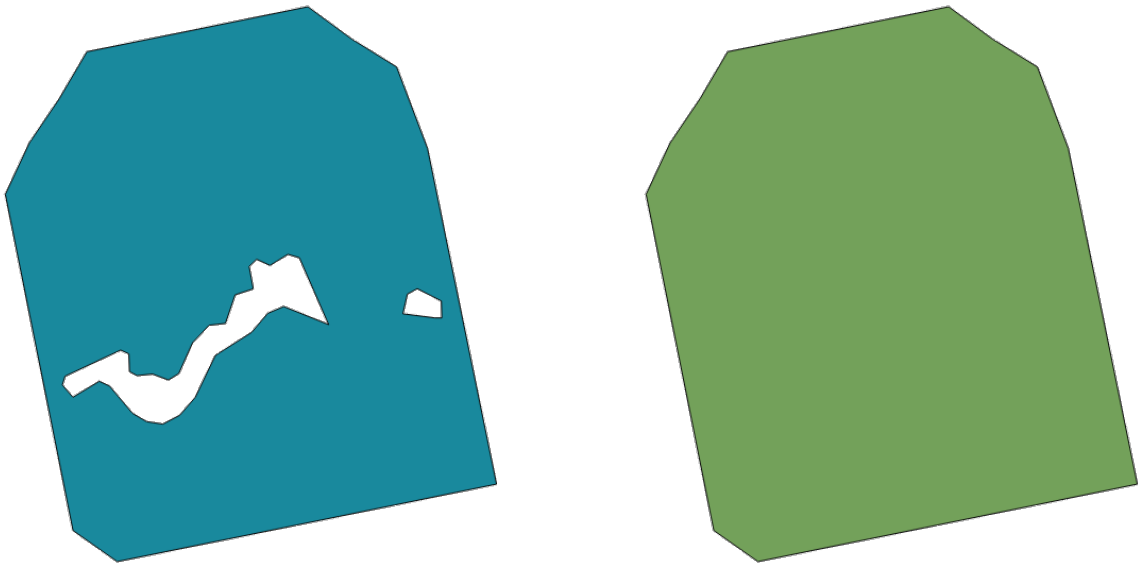
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Delete holes

Takes a polygon layer and removes holes in polygons. It creates a new vector layer in which polygons with holes have been replaced by polygons with only their external ring. Attributes are not modified.

An optional minimum area parameter allows removing only holes which are smaller than a specified area threshold. Leaving this parameter at `0.0` results in all holes being removed.



24.106 Fig. : Before and after the cleaning

Leidžia *vietoje keisti* poligonų geoobjektus

Taip pat žiūrėkite

qgisfixgeometryhole

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Įvesties poligonų vektorinis sluoksnis
Remove holes with area less than Pasirinktinis	`MIN_AREA`	[skaičius: slankaus kablelio] Numatytas: 0.0	Only holes with an area less than this threshold will be deleted. With a value of `0.0`, **all** the holes will be deleted.
Cleaned	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Cleaned	`OUTPUT`	[toks pat, kaip įvestis]	The output (cleaned) vector layer

Pythono kodas

Algorithm ID: `native:deleteholes`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

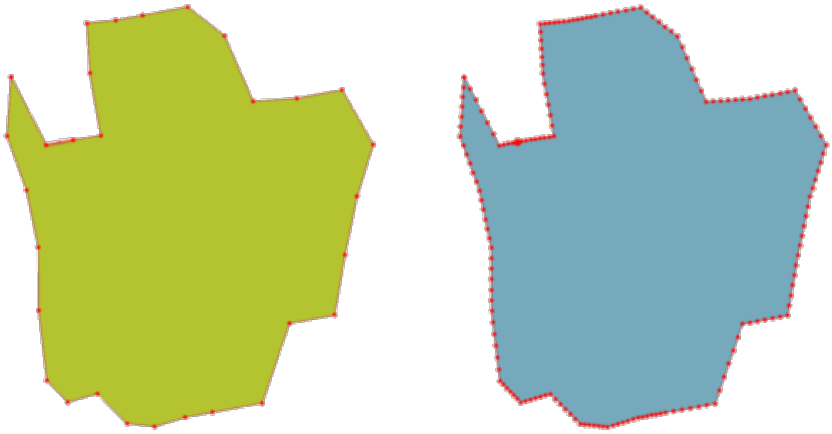
algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Densify by count

Takes a polygon or line layer and generates a new one in which the geometries have a larger number of vertices than the original one.

If the geometries have Z or M values present then these will be linearly interpolated at the added vertices.

The number of new vertices to add to each segment is specified as an input parameter.



24.107 Fig. : Red points show the vertices before and after the densify

Allows *features in-place modification* of line and polygon features

Numatytas meniu: *Vektorius* ► *Geometrijos įrankiai*

Taip pat žiūrėkite

Densify by interval

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija, poligonas]	Įvesties linijų ar poligonų vektorinis sluoksnis
Vertices to add	`VERTICES`	[skaičius: sveikas skaičius] Numatytas: 1	Number of vertices to add to each segment
Densified	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Densified	`OUTPUT`	[toks pat, kaip įvestis]	The output (densified) vector layer

Pythono kodas

Algorithm ID: `native:densifygeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Densify by interval

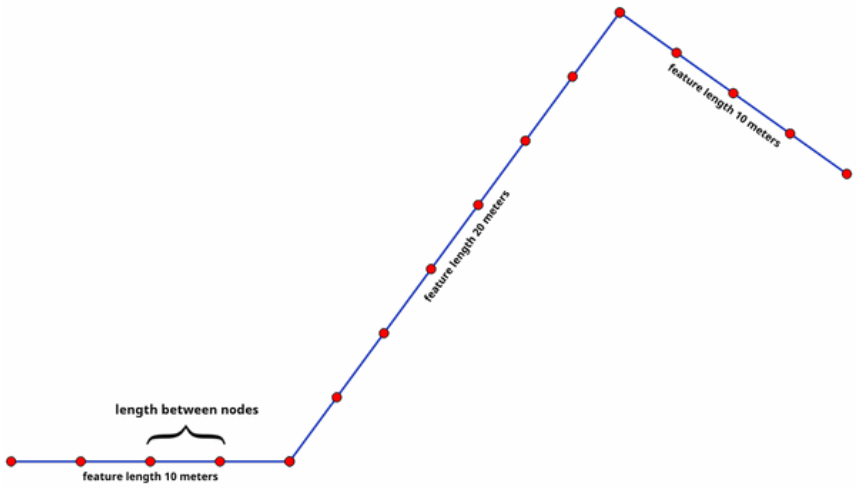
Takes a polygon or line layer and generates a new one in which the geometries have a larger number of vertices than the original one.

The geometries are densified by adding regularly placed extra vertices inside each segment so that the maximum distance between any two vertices does not exceed the specified distance.

If the geometries have Z or M values present then these will be linearly interpolated at the added vertices.

Pavyzdys

Specifying a distance of 3 would cause the segment `[0 0] -> [10 0]` to be converted to `[0 0] -> [2.5 0] -> [5 0] -> [7.5 0] -> [10 0]`, since 3 extra vertices are required on the segment and spacing these at 2.5 increments allows them to be evenly spaced over the segment.



24.108 Fig. : Densify geometry at a given interval

Allows *features in-place modification* of line and polygon features

Taip pat žiūrėkite

Densify by count

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija, poligonas]	Įvesties linijų ar poligonų vektorinis sluoksnis
Interval between vertices to add	`INTERVAL`	[skaičius: slankaus kablelio] Numatytas: 1.0	Maximum distance between two consecutive vertices
Densified	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Densified	`OUTPUT`	[toks pat, kaip įvestis]	The output (densified) vector layer

Pythono kodas

Algorithm ID: `native:densifygeometriesgivenaninterval`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

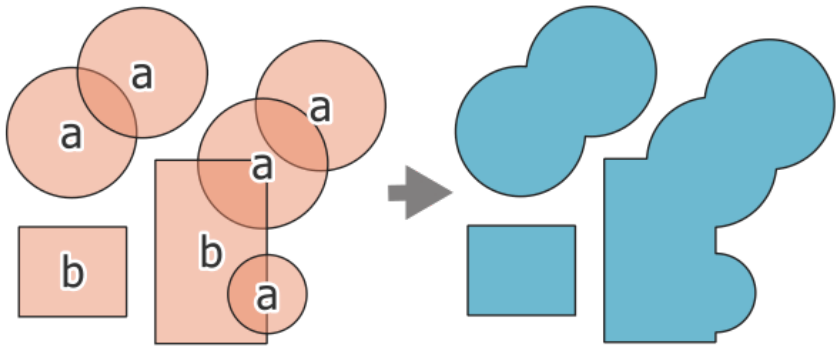
algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Dissolve

Takes a vector layer and combines its features into new features. One or more attributes can be specified to dissolve features belonging to the same class (having the same value for the specified attributes), alternatively all features can be dissolved to a single feature.

All output geometries will be converted to multi geometries. In case the input is a polygon layer, common boundaries of adjacent polygons being dissolved will get erased. If enabled, the optional „Keep disjoint features separate“ setting will cause features and parts that do not overlap or touch to be exported as separate features (instead of parts of a single multipart feature).

The resulting attribute table will have the same fields as the input layer. The values in the output layer's fields are the ones of the first input feature that happens to be processed.



24.109 Fig. : Dissolving a layer into a single multipart feature

Numatytas meniu: *Vektorius* ► *Geometrijos įrankiai*

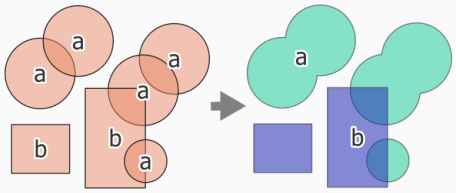
> **Įspėjimas**
>
> Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

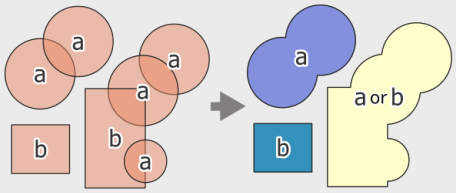
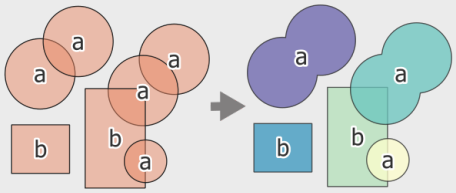
Sulieti padengimą, *Agreguoti*, *Collect geometries*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Dissolve field(s) Pasirinktinis	`FIELD`	[lentelės laukas: bet koks] [sąrašas] Default: []	Features having the same value for the selected field(s) will be replaced with a single one and their geometries are merged. If no field is provided then all the features are dissolved, resulting in a single (multipart) feature.  24.110 Fig. : Dissolve the polygon layer on a common attribute (2 multipart features)
Sulietas	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Palikti nesusijusius geoobjektus atskirais	`SEPARATE_DISJO`	[loginis] Numatytas: ne	Parts of dissolved features are exported as separate features (instead of parts of a multipart feature).  24.111 Fig. : source (left), dissolve all (3 distinct features - right)  24.112 Fig. : source (left), dissolve on field (5 distinct features - right)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sulietas	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer with dissolved geometries

Pythono kodas

Algorithm ID: `native:dissolve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Drape (set Z value from raster)

Uses values sampled from a band within a raster layer to set the Z value for every overlapping vertex in the feature geometry. The raster values can optionally be scaled by a preset amount.

If Z values already exist in the layer, they will be overwritten with the new value. If no Z values exist, the geometry will be upgraded to include the Z dimension.

Allows *features in-place modification* of point, line, and polygon features with Z enabled

Taip pat žiūrėkite

Set M value from raster, *Set Z value*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Rastro sluoksnis	`RASTER`	[rastras]	Raster layer with Z values
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	The raster band to take the Z values from
Value for NoData or non-intersecting vertices	`NODATA`	[skaičius: slankaus kablelio] Numatytas: 0.0	Value to use in case the vertex does not intersect (a valid pixel of) the raster
Scale factor	`SCALE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Scaling value: the band values are multiplied by this value.
Offset	`OFFSET`	[skaičius: slankaus kablelio] Numatytas: 0.0	Offset value: it is algebraically added to the band values after applying the „Scale factor“.
Updated	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer (with Z values from the raster layer). *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Updated	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer with Z values from the raster layer

Pythono kodas

Algorithm ID: `native:setzfromraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Drop M/Z values

Removes M (measure) or Z (altitude) values from input geometries.

Taip pat žiūrėkite

Set M value, *Set Z value*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Input vector layer with M or Z values
Drop M Values	`DROP_M_VALUES`	[loginis] Numatytas: ne	Removes the M values from the geometries
Drop Z Values	`DROP_Z_VALUES`	[loginis] Numatytas: ne	Removes the Z values from the geometries
Z/M Dropped	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Z/M Dropped	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer (identical to the input layer, except that the M and/or Z dimensions have been removed from the geometries).

Pythono kodas

Algorithm ID: `native:dropmzvalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Eliminate selected polygons

Combines selected polygons of the input layer with certain adjacent polygons by erasing their common boundary. The adjacent polygon can be either the one with the largest or smallest area or the one sharing the largest common boundary with the polygon to be eliminated.

Eliminate is normally used to get rid of sliver polygons, i.e. tiny polygons that are a result of polygon intersection processes where boundaries of the inputs are similar but not identical.

Numatytas meniu: *Vektorius ► Geometrijos įrankiai*

Taip pat žiūrėkite

Fix geometries

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Įvesties poligonų vektorinis sluoksnis
Merge selection with the neighboring polygon with the	`MODE`	[sąrašas] Numatytas: jokio	Choose the parameter to use in order to get rid of the selected polygons: • 0 — Largest Area • 1 — Smallest Area • 2 — Largest Common Boundary
Eliminated	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Eliminated	`OUTPUT`	[vektorius: poligonas]	The output polygon vector layer.

Pythono kodas

Algorithm ID: `qgis:eliminateselectedpolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Explode lines

Takes a lines layer and creates a new one in which each line layer is replaced by a set of lines representing the segments in the original line.

Each line in the resulting layer contains only a start and an end point, with no intermediate vertices between them.

If the input layer consists of CircularStrings or CompoundCurves, the output layer will be of the same type and contain only single curve segments.

Pastaba

- Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.
- This algorithm does not require valid geometries as input.

24.113 Fig. : The original line layer and the exploded one

Leidžia linijų *geoobjektus keisti vietoje*

Taip pat žiūrėkite

Subdivide, *Line substring*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Exploded	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Exploded	`OUTPUT`	[vektorius: linija]	The output line vector layer with features representing each segment of the input layer.

Pythono kodas

Algorithm ID: `native:explodelines`

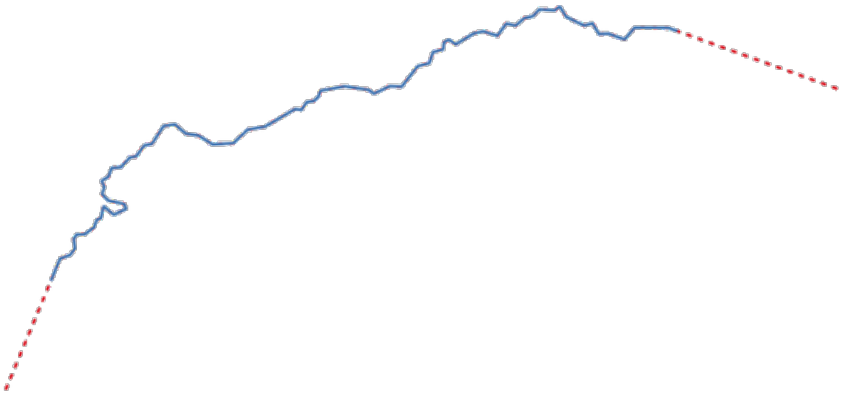
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Extend lines

Extends line geometry by a specified amount at the start and end of the line.

Lines are extended using the bearing of the first and last segment in the line.



24.114 Fig. : The red dashes represent the initial and final extension of the original layer

Leidžia linijų *geoobjektus keisti vietoje*

Taip pat žiūrėkite

Line substring

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Start distance	`START_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Distance by which to extend the first segment of the line (starting point)
End distance	`END_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Distance by which to extend the last segment of the line (ending point)
Extended	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Extended	`OUTPUT`	[vektorius: linija]	The output (extended) line vector layer.

Pythono kodas

Algorithm ID: `native:extendlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Extract M values

Extracts M values from geometries into feature attributes.

By default only the M value from the first vertex of each feature is extracted, however the algorithm can optionally calculate statistics on all of the geometry's M values, including sum, mean, minimum and maximum.

Taip pat žiūrėkite

Extract Z values, *Set M value*, *Drop M/Z values*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis

continues on next page

Table 24.650 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Summaries to calculate	SUMMARIES	[sąrašas] Numatytas: [0]	Statistics on the M values of a geometry. One or more of: • 0 — First • 1 — Last • 2 — Count • 3 — Sum • 4 — Mean • 5 — Median • 6 — St.dev (pop) • 7 — Minimum • 8 — Maximum • 9 — Range • 10 — Minority • 11 — Majority • 12 — Variety • 13 — Q1 • 14 — Q3 • 15 — IQR
Output column prefix	COLUMN_PREFIX	[tekstas] Default: ‚m_‘	The prefix for the output (M) column
Extracted	OUTPUT	[toks pat, kaip įvestis] Numatytas: [Kurti laikiną sluoksnį]	Nurodykite išvesties sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (TEMPORARY_OUTPUT) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Extracted	OUTPUT	[toks pat, kaip įvestis]	The output vector layer (with M values)

Pythono kodas

Algorithm ID: native:extractmvalues

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Extract specific vertices

Takes a vector layer and generates a point layer with points representing specific vertices in the input geometries.

For instance, this algorithm can be used to extract the first or last vertices in the geometry. The attributes associated to each point are the same ones associated to the feature that the vertex belongs to.

The vertex indices parameter accepts a comma separated string specifying the indices of the vertices to extract. The first vertex corresponds to an index of 0, the second vertex has an index of 1, etc. Negative indices can be used to find vertices at the end of the geometry, e.g., an index of -1 corresponds to the last vertex, -2 corresponds to the second last vertex, etc.

Additional fields are added to the vertices indicating the specific vertex position (e.g., 0, -1, etc), the original vertex index, the vertex's part and its index within the part (as well as its ring for polygons), distance along the original geometry and bisector angle of vertex for the original geometry.

Leidžia taškų *geoobjektus keisti vietoje*

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Extract vertices, *Filter vertices by M value*, *Filter vertices by Z value*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Vertex indices	`VERTICES`	[tekstas] Default: ‚0‘	Comma-separated string of the indices of the vertices to extract.
Vertices	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Vertices	`OUTPUT`	[vektorius: taškas]	The output (point) vector layer containing the specified vertices from the input layer geometries. Other than the input attributes, the output layer also contains the following fields: • `vertex_pos`: the index of the vertex in the geometry, as requested by the user. • `vertex_index`: the index of the vertex within the geometry. The first vertex is indexed as 0. • `vertex_part`: the part number of the geometry that the vertex belongs to. For single-part geometries, this is always 0. For multi-part geometries (e.g. MultiPolygon or MultiLine), this indicates which part the vertex comes from. • `vertex_part_ring`: for polygon geometries, this indicates whether the vertex belongs to the outer ring (0) or one of the inner rings (1, 2, …). • `vertex_part_index`: the index of the vertex in its specific ring (outer or inner). • `distance`: the cumulative length of line segments from the first vertex of the geometry. • `angle`: the angle (in degrees) between the previous and the current vertex.

Pythono kodas

Algorithm ID: `native:extractspecificvertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

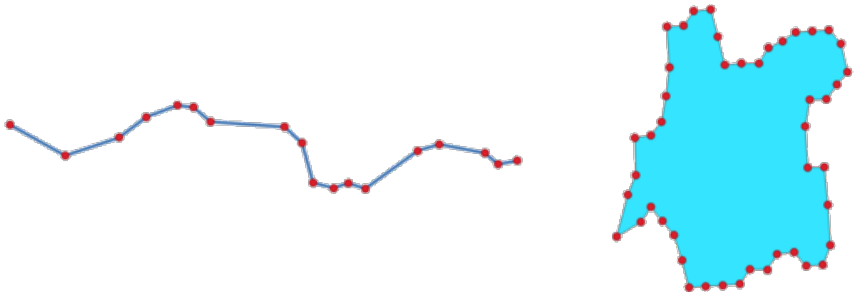
algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Extract vertices

Takes a vector layer and generates a point layer with points representing the vertices in the input geometries.

The attributes associated to each point are the same ones associated to the feature that the vertex belongs to.

Additional fields are added to the vertices indicating the vertex index (beginning at 0), the feature's part and its index within the part (as well as its ring for polygons), distance along original geometry and bisector angle of vertex for original geometry.



24.115 Fig. : Vertices extracted for line and polygon layer

Leidžia taškų *geoobjektus keisti vietoje*

Numatytas meniu: *Vektorius ► Geometrijos įrankiai*

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Extract specific vertices, *Filter vertices by M value*, *Filter vertices by Z value*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Vertices	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Vertices	`OUTPUT`	[vektorius: taškas]	The output (point) vector layer containing the vertices from the input layer geometries. Other than the input attributes, the output layer also contains the following fields: • `vertex_index`: the index of the vertex within the geometry. The first vertex is indexed as 0. • `vertex_part`: the part number of the geometry that the vertex belongs to. For single-part geometries, this is always 0. For multi-part geometries (e.g. MultiPolygon or MultiLine), this indicates which part the vertex comes from. • `vertex_part_ring`: for polygon geometries, this indicates whether the vertex belongs to the outer ring (0) or one of the inner rings (1, 2, …). • `vertex_part_index`: the index of the vertex in its specific ring (outer or inner). • `distance`: the distance from the previous vertex. • `angle`: returns approximate angle at a vertex. Usually the average angle between adjacent segments, and can be pictured as the orientation of a line following the curvature of the geometry at the specified vertex.

Pythono kodas

Algorithm ID: `native:extractvertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Extract Z values

Extracts Z values from geometries into feature attributes.

By default only the Z value from the first vertex of each feature is extracted, however the algorithm can optionally calculate statistics on all of the geometry's Z values, including sum, mean, minimum and maximum.

Taip pat žiūrėkite

Extract M values, *Set Z value*, *Drop M/Z values*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Summaries to calculate	`SUMMARIES`	[sąrašas] Numatytas: [0]	Statistics on the Z values of a geometry. One or more of: • 0 — First • 1 — Last • 2 — Count • 3 — Sum • 4 — Mean • 5 — Median • 6 — St.dev (pop) • 7 — Minimum • 8 — Maximum • 9 — Range • 10 — Minority • 11 — Majority • 12 — Variety • 13 — Q1 • 14 — Q3 • 15 — IQR
Output column prefix	`COLUMN_PREFIX`	[tekstas] Default: ‚z_‘	The prefix for the output (Z) column
Extracted	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Extracted	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer (with Z values)

Pythono kodas

Algorithm ID: `native:extractzvalues`

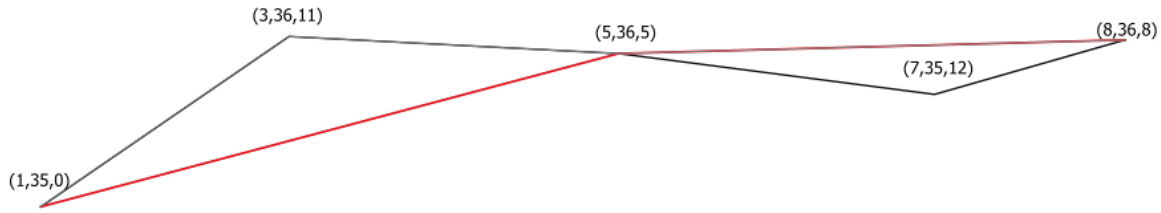
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Filter vertices by M value

Filters away vertices based on their M value, returning geometries with only vertex points that have a M value greater than or equal to the specified minimum value and/or less than or equal to the maximum value.

If the minimum value is not specified then only the maximum value is tested, and similarly if the maximum value is not specified then only the minimum value is tested.



24.116 Fig. : The red line represents the black line with only vertices whose M value is <=10.

Allows *features in-place modification* of point, line and polygon features with M enabled

Įspėjimas

Depending on the input geometry attributes and the filters used, the geometries created by this algorithm may no longer be valid.

Taip pat žiūrėkite

Filter vertices by Z value, *Extract vertices*, *Extract specific vertices*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Input vector layer to remove vertices from
Minimum Pasirinktinis	`MIN`	[skaičius: slankaus kablelio] Numatytas: jokio	Minimum of M values allowed
Maximum Pasirinktinis	`MAX`	[skaičius: slankaus kablelio] Numatytas: jokio	Maximum of M values allowed
Filtered	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Filtered	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer of features with only the filtered vertices.

Pythono kodas

Algorithm ID: `native:filterverticesbym`

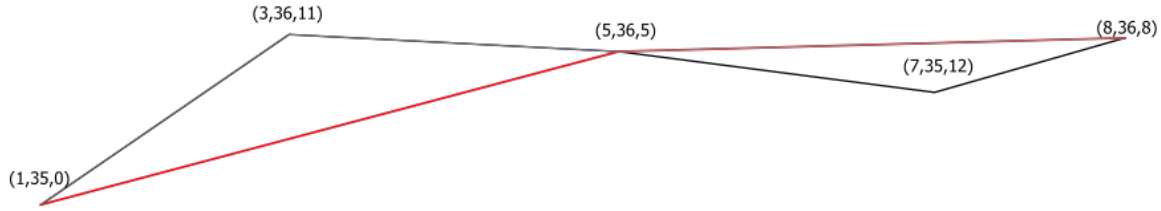
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Filter vertices by Z value

Filters away vertices based on their Z value, returning geometries with only vertex points that have a Z value greater than or equal to the specified minimum value and/or less than or equal to the maximum value.

If the minimum value is not specified then only the maximum value is tested, and similarly if the maximum value is not specified then only the minimum value is tested.



24.117 Fig. : The red line represents the black line with only vertices whose Z value is <=10.

Allows *features in-place modification* of point, line and polygon features with Z enabled

Įspėjimas

Depending on the input geometry attributes and the filters used, the geometries created by this algorithm may no longer be valid. You may need to run the *Fix geometries* algorithm to ensure their validity.

Taip pat žiūrėkite

Filter vertices by M value, *Extract vertices*, *Extract specific vertices*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Input vector layer to remove vertices from
Minimum Pasirinktinis	`MIN`	[skaičius: slankaus kablelio] Numatytas: jokio	Minimum of Z values allowed
Maximum Pasirinktinis	`MAX`	[skaičius: slankaus kablelio] Numatytas: jokio	Maximum of Z values allowed
Filtered	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Filtered	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer of features with only the filtered vertices.

Pythono kodas

Algorithm ID: `native:filterverticesbyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Fix geometries

Attempts to create a valid representation of a given invalid geometry without losing any of the input vertices. Already valid geometries are returned without further intervention. Always outputs multi-geometry layer.

Allows *features in-place modification* of point, line, and polygon features without M enabled

Įspėjimas

M values will be dropped from the output.

Taip pat žiūrėkite

Tikrinti tvarkingumą, *Fix Geometry*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis

continues on next page

Table 24.662 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Repair method	`METHOD`	[sąrašas] Numatytas: 1	Method used to repair the geometries. One of: • 0 — `Linework`: combines all rings into a set of noded lines and then extracts valid polygons from that linework • 1 — `Structure`: first makes all rings valid and then merges shells and subtracts holes from shells to generate valid result. Assumes that holes and shells are correctly categorized. **Dėmesio** Using this method requires QGIS installed with GEOS >= 3.10 (see *Help* ► *About* menu).
Fixed geometries	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Fixed geometries	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer with fixed geometries.

Pythono kodas

Algorithm ID: `native:fixgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Force right-hand-rule

Forces polygon geometries to respect the Right-Hand-Rule, in which the area that is bounded by a polygon is to the right of the boundary. In particular, the exterior ring is oriented in a clockwise direction and any interior rings in a counter-clockwise direction.

Leidžia *vietoje keisti* poligonų geoobjektus

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Įvesties vektorinis sluoksnis
Reoriented	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Reoriented	`OUTPUT`	[vektorius: poligonas]	The output vector layer with reoriented geometries.

Pythono kodas

Algorithm ID: `native:forcerhr`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Geodesic line split at antimeridian

Splits a line into multiple geodesic segments, whenever the line crosses the antimeridian (±180 degrees longitude).

Splitting at the antimeridian helps the visual display of the lines in some projections. The returned geometry will always be a multi-part geometry.

Whenever line segments in the input geometry cross the antimeridian, they will be split into two segments, with the latitude of the breakpoint being determined using a geodesic line connecting the points either side of this segment. The current project ellipsoid setting will be used when calculating this breakpoint.

If the input geometry contains M or Z values, these will be linearly interpolated for the new vertices created at the antimeridian.

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.

Leidžia linijų *geoobjektus keisti vietoje*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Split	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output line vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Split	`OUTPUT`	[vektorius: linija]	The output line vector layer split at the antimeridian.

Pythono kodas

Algorithm ID: `native:antimeridiansplit`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Geometry by expression

Updates existing geometries (or creates new geometries) for input features by use of a QGIS expression.

This allows complex geometry modifications which can utilize all the flexibility of the QGIS expression engine to manipulate and create geometries for output features.

For help with QGIS expression functions, see the inbuilt help available in the *expression builder*.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties vektorinis sluoksnis
Output geometry type	`OUTPUT_GEOMETR`	[sąrašas] Numatytas: 0	The output geometry strongly depends on the expression: for instance, if you create a buffer the geometry type has to be polygon. One of: • 0 — Polygon • 1 — Linija • 2 — Point
Output geometry has z values	`WITH_Z`	[loginis] Numatytas: ne	Choose if the output geometry should include the Z dimension
Output geometry has m values	`WITH_M`	[loginis] Numatytas: ne	Choose if the output geometry should include the M dimension
Geometry expression	`EXPRESSION`	[išraiška] Default: ‚$geometry‘	Add the geometry expression you want to use. You can use the button to open the Expression Dialog. The dialog lists all the relevant expressions, together with their help and guide.
Modified geometry	`OUTPUT`	[vektorius: geometrija] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Modified geometry	`OUTPUT`	[vektorius: geometrija]	Išvesties vektorinis sluoksnis

Pythono kodas

Algorithm ID: `native:geometrybyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

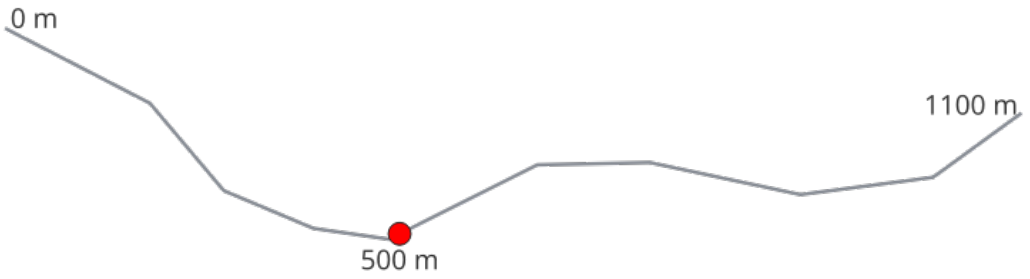
Interpolate point on line

Creates a point geometry interpolated at a set distance along line or curve geometries.

Z and M values are linearly interpolated from existing values.

If a multipart geometry is encountered, only the first part is considered when calculating the substring.

If the specified distance is greater than the input feature's length, the resultant feature will have a null geometry.



24.118 Fig. : Interpolated point at 500m of the beginning of the line

Taip pat žiūrėkite

Points along geometry

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija, poligonas]	Įvesties linijų ar poligonų vektorinis sluoksnis
Atstumas	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Distance from the beginning of the line
Interpolated points	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Interpolated points	`OUTPUT`	[vektorius: taškas]	The output point vector layer with features at a set distance along the line or polygon boundary

Pythono kodas

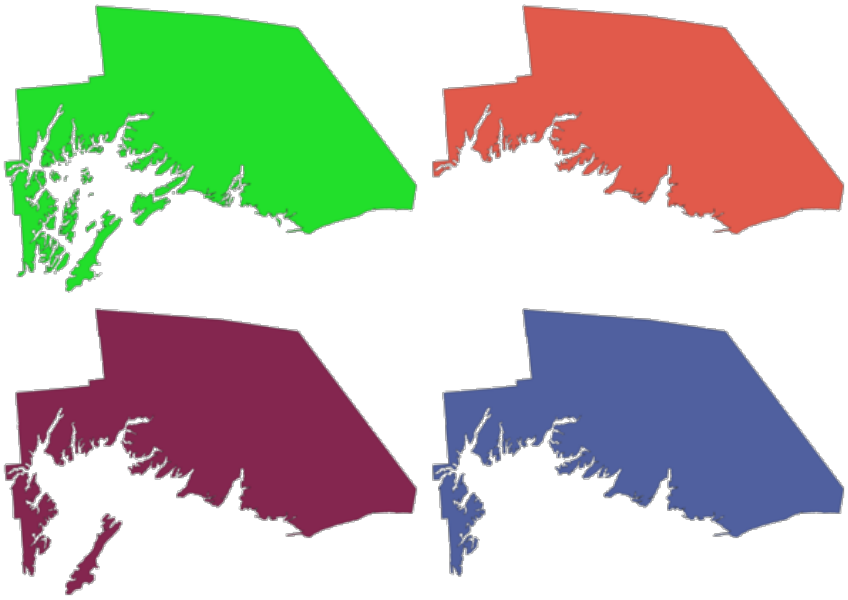
Algorithm ID: `native:interpolatepoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Keep N biggest parts

Takes a layer with polygons or multipolygons and returns a new layer in which only the *n* largest polygons of each multipolygon feature are kept. If a feature has *n* or fewer parts, the feature will just be copied.



24.119 Fig. : Clockwise from top left: original multipart feature, one, two and three biggest parts kept

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Poligonai	`INPUT`	[vektorius: poligonas]	Įvesties poligonų vektorinis sluoksnis
Parts to keep	`PARTS`	[skaičius: sveikas skaičius] Numatytas: 1	Number of parts to keep. If 1, only the biggest part of the feature will be kept.
Parts	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output polygon vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Parts	`OUTPUT`	[vektorius: poligonas]	The output polygon vector layer with the N biggest parts of each feature

Pythono kodas

Algorithm ID: `native:keepnbiggestparts`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Line substring

Returns the portion of a line (or curve) which falls between the specified start and end distances (measured from the beginning of the line).

Z and M values are linearly interpolated from existing values.

If a multipart geometry is encountered, only the first part is considered when calculating the substring.

24.120 Fig. : Substring line with starting distance set at 0 meters and the ending distance at 250 meters.

Leidžia linijų *geoobjektus keisti vietoje*

Taip pat žiūrėkite

Extend lines

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Start distance	`START_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Distance along the input line to the start point of the output feature
End distance	`END_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Distance along the input line to the end point of the output feature
Substring	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output line vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Substring	`OUTPUT`	[vektorius: linija]	The output line vector layer.

Pythono kodas

Algorithm ID: `native:linesubstring`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Lines to polygons

Generates a polygon layer using as polygon rings the lines from an input line layer.

The attribute table of the output layer is the same as the one of the input layer.

Numatytas meniu: *Vektorius* ► *Geometrijos įrankiai*

Taip pat žiūrėkite

Polygons to lines, *Polygonize*, *Convert geometry type*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Poligonai	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output polygon vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Poligonai	`OUTPUT`	[vektorius: poligonas]	The output polygon vector layer.

Pythono kodas

Algorithm ID: `qgis:linestopolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Merge lines

Joins all connected parts of MultiLineString geometries into single LineString geometries.

If any parts of the input MultiLineString geometries are not connected, the resultant geometry will be a MultiLineString containing any lines which could be merged and any non-connected line parts.

Leidžia linijų *geoobjektus keisti vietoje*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Sulietas	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output line vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sulietas	`OUTPUT`	[vektorius: linija]	The output (merged) line vector layer.

Pythono kodas

Algorithm ID: `native:mergelines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

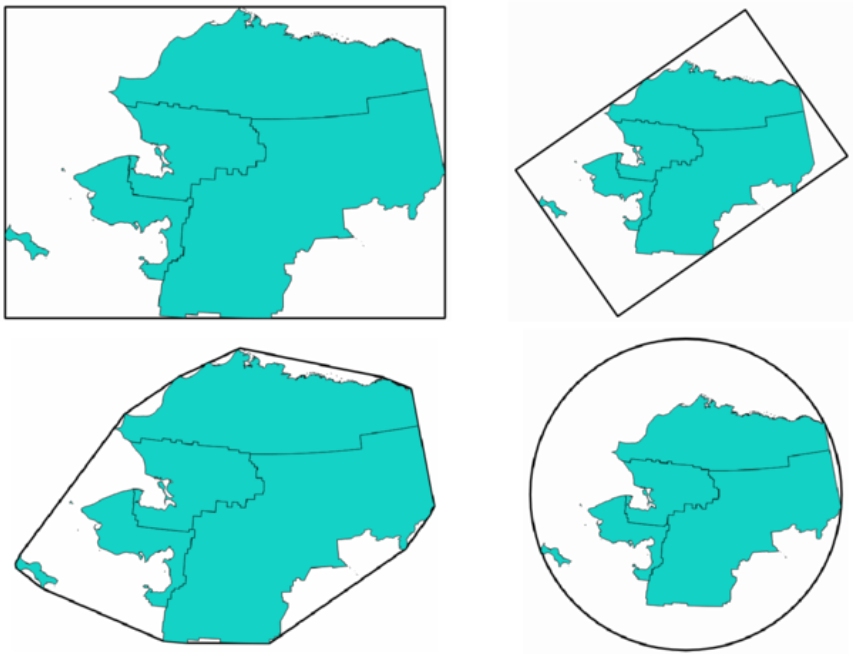
algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Minimum bounding geometry

Creates geometries which enclose the features from an input layer. The features can be grouped by a field. The output layer will then contain one feature per group value with a geometry (MBB) that covers the geometries of the features with matching value.

The following enclosing geometry types are supported:

- bounding box (envelope)
- oriented rectangle
- circle
- convex hull



24.121 Fig. : Clockwise from top left: envelope, oriented rectangle, circle, convex hull

Taip pat žiūrėkite

Minimum enclosing circles

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Field Pasirinktinis	`FIELD`	[lentelės laukas: bet koks]	Features can be grouped by a field. If set, this causes the output layer to contain one feature per grouped value with a minimal geometry covering only the features with matching values.

continues on next page

Table 24.680 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Geometry type	`TYPE`	[sąrašas] Numatytas: 0	Enclosing geometry types. One of: • 0 — Envelope (Bounding Box) • 1 — Minimum Oriented Rectangle • 2 — Minimum Enclosing Circle • 3 — Convex Hull
Bounding geometry	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output polygon vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Bounding geometry	`OUTPUT`	[vektorius: poligonas]	The output (bounding) polygon vector layer. It has the fields from the input layer and, depending on the selected parameter `TYPE`, the output also has the attributes: • `width`, `height`, `area` and `perimeter` of the Envelope (Bounding Box) • `width`, `height`, `area`, `angle` and `perimeter` of the Minimum Oriented Rectangle • `radius` and `area` of the Minimum Enclosing Circle • `area` and `perimeter` of the Convex Hull

Pythono kodas

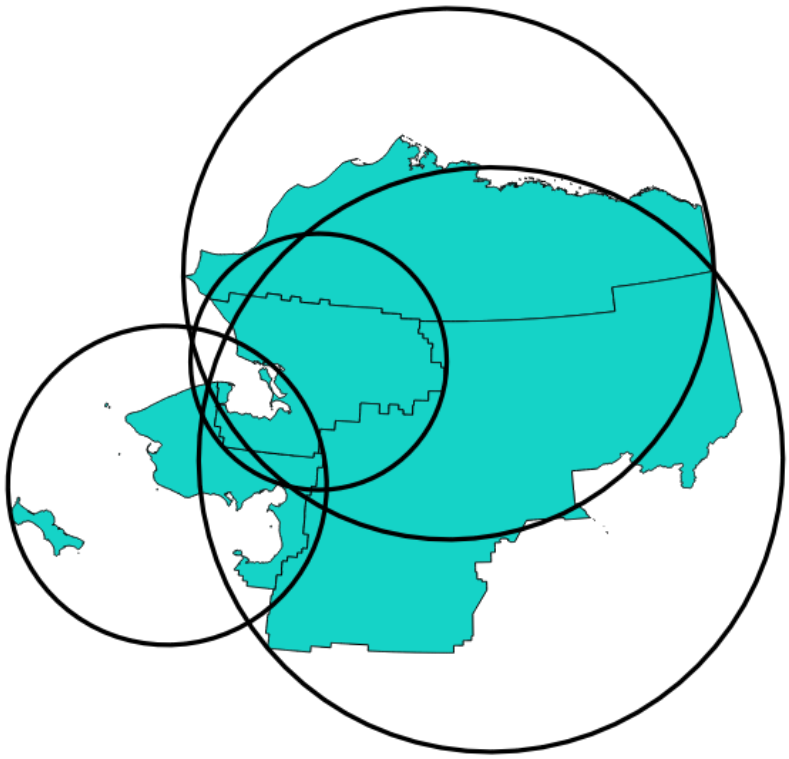
Algorithm ID: `qgis:minimumboundinggeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Minimum enclosing circles

Calculates the minimum enclosing circles of the features in the input layer.



24.122 Fig. : Enclosing circles for each feature

Leidžia *vietoje keisti* poligonų geoobjektus

Taip pat žiūrėkite

Minimum bounding geometry

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Number of segments in circles	`SEGMENTS`	[skaičius: sveikas skaičius] Default: 72	The number of segments used to approximate a circle. Minimum 8, maximum 100000.

continues on next page

Table 24.682 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Minimum enclosing circles	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output polygon vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Minimum enclosing circles	`OUTPUT`	[vektorius: poligonas]	The output polygon vector layer. Other than the input attributes, the output layer also contains the following fields: `area` and `radius` of the generated polygon.

Pythono kodas

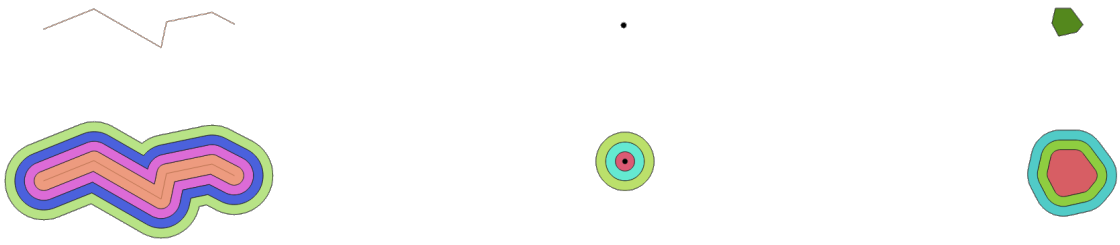
Algorithm ID: `native:minimumenclosingcircle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Multi-ring buffer (constant distance)

Computes multi-ring (*donut*) buffer for the features of the input layer, using a fixed or dynamic distance and number of rings.



24.123 Fig. : Multi-ring buffer for a line, point and polygon layer

Leidžia *vietoje keisti* poligonų geoobjektus

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Buferis, *Variable distance buffer*, *Rectangles, ovals, diamonds*, *Single sided buffer*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Number of rings	`RINGS`	[skaičius: sveikas skaičius] Numatytas: 1	The number of rings. It can be a unique value (same number of rings for all the features) or it can be taken from features data (the number of rings depends on feature values).
Distance between rings	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Distance between the rings. It can be a unique value (same distance for all the features) or it can be taken from features data (the distance depends on feature values).
Multi-ring buffer (constant distance)	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output polygon vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Multi-ring buffer (constant distance)	`OUTPUT`	[vektorius: poligonas]	The output polygon vector layer. Other than the input attributes, the output layer also contains the following fields: `ringId` and `distance` of the generated buffer polygon.

Pythono kodas

Algorithm ID: `native:multiringconstantbuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Multipart to singleparts

Splits multipart features in the input layer into singlepart features.

The attributes of the output layer are the same as the original ones but divided into single features.

24.124 Fig. : Left the multipart source layer and right the single part output result

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Numatytas meniu: *Vektorius* ► *Geometrijos įrankiai*

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Collect geometries, *Promote to multipart*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Single parts	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output polygon vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Single parts	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer.

Pythono kodas

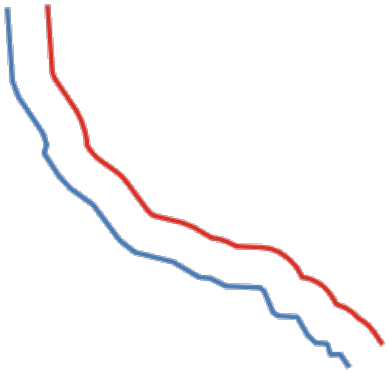
Algorithm ID: `native:multiparttosingleparts`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Offset lines

Offsets lines by a specified distance. Positive distances will offset lines to the left, and negative distances will offset them to the right.



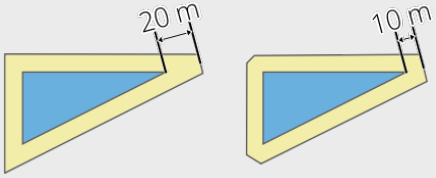
24.125 Fig. : Mėlynas - pradinis sluoksnis, raudonas - paslinktasis

Leidžia linijų *geoobjektus keisti vietoje*

Taip pat žiūrėkite

Paslinktų (lygiagrečių) linijų masyvas, *Translate*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Atstumas	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 10.0	Offset distance. You can use the Data Defined button on the right to choose a field from which the radius will be calculated. This way you can have different radius for each feature (see *Variable distance buffer*).
Segmentai	`SEGMENTS`	[skaičius: sveikas skaičius] Numatyta: 8	Valdo skaičių linijų segmentų, kurie naudojami aproksimuojant apskritimo ketvirčius kuriant užapvalintus poslinkius.
Jungimo stilius	`JOIN_STYLE`	[sąrašas] Numatytas: 0	Nurodo, ar paslinkus linijos kampus reikia naudoti apvalias, mitros ar kampines jungtis. Parinktys yra: • 0 — Apvalūs • 1 — Mitra • 2 — Kampas 24.126 Fig. : Apvalūs, mitros ir kampo jungimo stiliai
Mitros riba	`MITER_LIMIT`	[skaičius: slankaus kablelio] Numatyta: 2.0	Nustato maksimalų atstumą nuo paslenkamos geometrijos kuriant mitros jungtį. Nurodomas kaip poslinkio atstumo koeficientas (taikoma tik mitros jungimo stiliui). Minimalus: 1.0  24.127 Fig. : 10m buferis su 2 riba ir 10m buferis su 1 riba
Offset	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output (offset) layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Offset	`OUTPUT`	[vektorius: linija]	Output (offset) line layer

Pythono kodas

Algorithm ID: `native:offsetline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

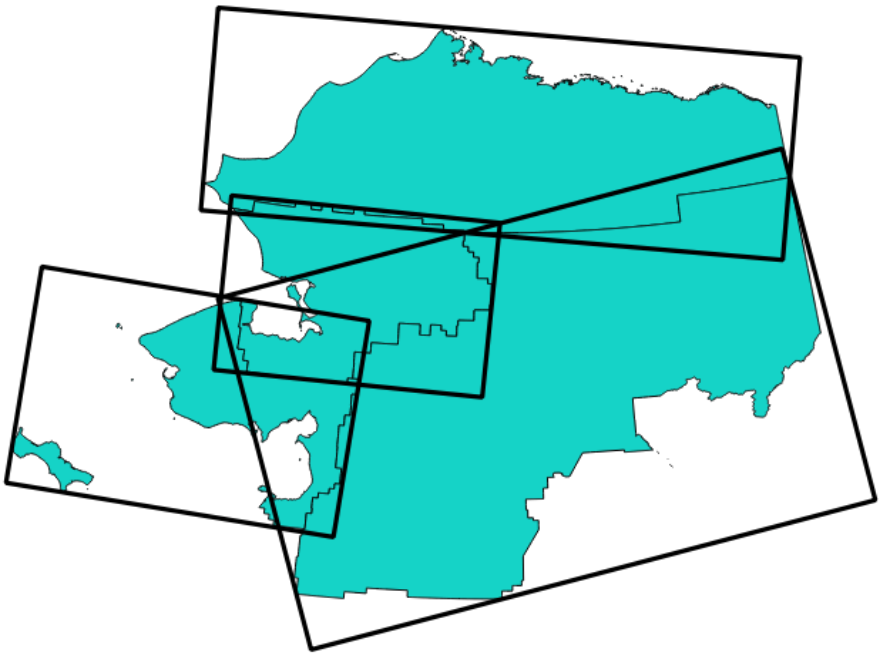
algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Oriented minimum bounding box

Calculates the minimum area rotated rectangle for each feature in the input layer.

Pastaba

For singlepart point features, the output corresponds to the bounding box of the geometry.



24.128 Fig. : Oriented minimum bounding box

Leidžia *vietoje keisti* poligonų geoobjektus

Taip pat žiūrėkite

Minimum bounding geometry

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Bounding boxes	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output polygon vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Bounding boxes	`OUTPUT`	[vektorius: poligonas]	The output polygon vector layer. Other than the input attributes, the output layer also contains the following fields: `width`, `height`, `angle`, `area` and `perimeter` of the generated polygon.

Pythono kodas

Algorithm ID: `native:orientedminimumboundingbox`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Orthogonalize

Attempts to orthogonalize the geometries of the input line or polygon layer. This process shifts the vertices in the geometries to try to make every angle in the geometry either a right angle or a straight line.

24.129 Fig. : In blue the source layer and in the red orthogonalized result

Allows *features in-place modification* of line and polygon features

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija, poligonas]	Įvesties linijų ar poligonų vektorinis sluoksnis
Maximum angle tolerance (degrees)	`ANGLE_TOLERANC`	[skaičius: slankaus kablelio] Default: 15.0	Specify the maximum deviation from a right angle or straight line a vertex can have for it to be adjusted. Smaller tolerances mean that only vertices which are already closer to right angles will be adjusted, and larger tolerances mean that vertices which deviate further from right angles will also be adjusted.
Maximum algorithm iterations	`MAX_ITERATIONS`	[skaičius: sveikas skaičius] Numatytas: 1000	Setting a larger number for the maximum number of iterations will result in a more orthogonal geometry at the cost of extra processing time.
Orthogonalized	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output polygon vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

continues on next page

Table 24.692 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Orthogonalized	`OUTPUT`	[toks pat, kaip įvestis]	The output polygon vector layer with adjusted angles.

Pythono kodas

Algorithm ID: `native:orthogonalize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Point on Surface

For each feature of the input layer, returns a point that is guaranteed to lie on the surface of the feature geometry.

Leidžia taškų *geoobjektus keisti vietoje*

Įspėjimas

Šis algoritmas gali išmesti esamus pirminius raktus ar FID reikšmes ir perkurti juos išvesties sluoksniuose, priklausomai nuo to, kokie įvesties parametrai.

Taip pat žiūrėkite

Centroidai

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Create point on surface for each part	`ANGLE_TOLERANC`	[loginis]	If checked, a point will be created for each part of the geometry.

continues on next page

Table 24.694 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Point	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output point vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Point	`OUTPUT`	[vektorius: taškas]	The output point vector layer.

Pythono kodas

Algorithm ID: `native:pointonsurface`

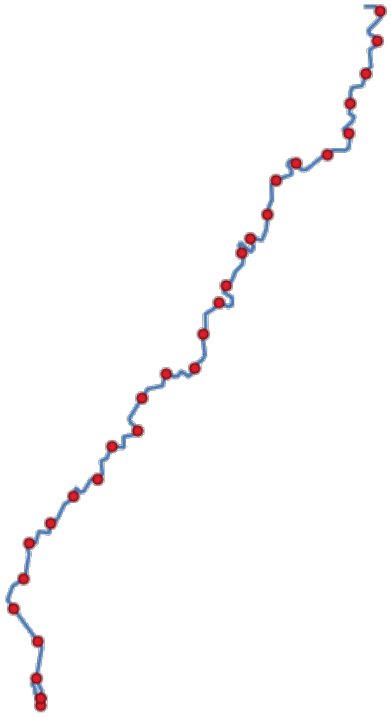
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Points along geometry

Creates points at regular intervals along line or polygon geometries. Created points will have new attributes added for the distance along the geometry and the angle of the line at the point.

An optional start and end offset can be specified, which controls how far from the start and end of the geometry the points should be created.



24.130 Fig. : Points created along the source line layer

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Interpolate point on line

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija, poligonas]	Įvesties linijų ar poligonų vektorinis sluoksnis
Atstumas	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Distance between two consecutive points along the line
Start offset	`START_OFFSET`	[skaičius: slankaus kablelio] Numatytas: 0.0	Distance from the beginning of the input line, representing the position of the first point.
End offset	`END_OFFSET`	[skaičius: slankaus kablelio] Numatytas: 0.0	Distance from the end of the input line, representing the position beyond which no point feature shoud be created.

continues on next page

Table 24.696 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Interpolated points	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Interpolated points	`OUTPUT`	[vektorius: taškas]	Point vector layer with features placed along lines or polygon boundaries of the input layer. Other than the input attributes on every related point, the output layer also contains the following fields: `distance` and `angle` of the generated point.

Pythono kodas

Algorithm ID: `native:pointsalonglines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Points displacement

Given a distance of proximity, identifies nearby point features and radially distributes them over a circle whose center represents their barycenter. A convenient tool to scatter overlaid features.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: taškas]	Input point vector layer
Minimum distance to other points	`PROXIMITY`	[skaičius: slankaus kablelio] Numatytas: 1.0	Distance below which point features are considered close. Close features are distributed altogether.
Displacement distance	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Radius of the circle on which close features are placed

continues on next page

Table 24.698 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Horizontal distribution for two point case	`HORIZONTAL`	[loginis] Numatytas: ne	When only two points are identified as close, aligns them horizontally on the circle instead of vertically.
Displaced	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Displaced	`OUTPUT`	[vektorius: taškas]	Output point vector layer

Pythono kodas

Algorithm ID: `qgis:pointsdisplacement`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Pole of inaccessibility

Calculates the pole of inaccessibility for a polygon layer, which is the most distant internal point from the boundary of the surface.

This algorithm uses the ‚polylabel' algorithm (Vladimir Agafonkin, 2016), which is an iterative approach guaranteed to find the true pole of inaccessibility within a specified tolerance. A more precise tolerance (lower value) requires more iterations and will take longer to calculate.

24.131 Fig. : Pole of inaccessibility

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Įvesties vektorinis sluoksnis
Tolerancija	`TOLERANCE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Set the tolerance for the calculation
Point	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output polygon vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Point	`OUTPUT`	[vektorius: taškas]	The output point vector layer. It will contain a new attribute with the distance `dist_pole` from the calculated pole to the polygon boundary.

Pythono kodas

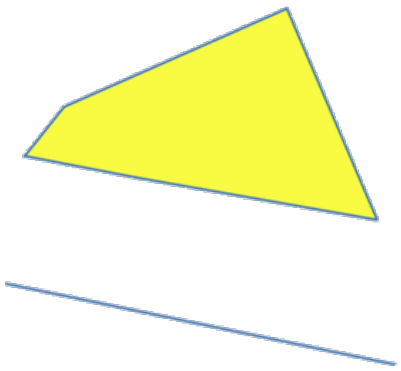
Algorithm ID: `native:poleofinaccessibility`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Polygonize

Creates a polygon layer whose features boundaries are generated from a line layer of **closed** features.



24.132 Fig. : The yellow polygons generated from the closed lines

Pastaba

The line layer must have closed shapes in order to be transformed into a polygon.

Taip pat žiūrėkite

Polygons to lines, *Lines to polygons*, *Convert geometry type*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Keep fields from the input layer	`KEEP_FIELDS`	[loginis] Numatytas: ne	Check to keep the fields (only the table structure, not the values) of the input layer

continues on next page

Table 24.702 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Polygons from lines	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output polygon vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Polygons from lines	`OUTPUT`	[vektorius: poligonas]	The output polygon vector layer from lines

Pythono kodas

Algorithm ID: `native:polygonize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Polygons to lines

Takes a polygon layer and creates a line layer, with lines representing the boundaries of the polygons in the input layer.

The attribute table of the output layer is the same as the one of the input layer.

24.133 Fig. : Black lines as the result of the algorithm

Numatytas meniu: *Vektorius ► Geometrijos įrankiai*

Taip pat žiūrėkite

Lines to polygons, *Polygonize*, *Convert geometry type*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Įvesties poligonų vektorinis sluoksnis
Lines	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output line vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Lines	`OUTPUT`	[vektorius: linija]	The output line vector layer from polygons

Pythono kodas

Algorithm ID: `native:polygonstolines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Project points (Cartesian)

Projects point geometries by a specified distance and bearing (azimuth).

Leidžia taškų *geoobjektus keisti vietoje*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: taškas]	Input point vector layer
Bearing (degrees from North)	`BEARING`	[skaičius: slankaus kablelio] Numatytas: 0.0	Clockwise angle starting from North, in degree (°) unit
Atstumas	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Distance to offset geometries, in layer units
Projected	`OUTPUT`	[vektorius: taškas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output point vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Projected	`OUTPUT`	[vektorius: taškas]	The output (projected) point vector layer

Pythono kodas

Algorithm ID: `native:projectpointcartesian`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Promote to multipart

Takes a vector layer with singlepart geometries and generates a new one in which all geometries are multipart.

Input features which are already multipart features will remain unchanged.

This algorithm can be used to force geometries to multipart types in order to be compatible with data providers that require multipart features.

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Taip pat žiūrėkite

Agreguoti, *Collect geometries*, *Convert to strictly multipart*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Multiparts	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output multipart vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Multiparts	`OUTPUT`	[toks pat, kaip įvestis]	The output multipart vector layer

Pythono kodas

Algorithm ID: `native:promotetomulti`

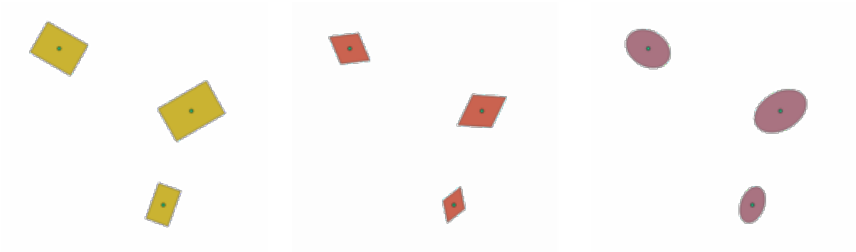
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Rectangles, ovals, diamonds

Creates a buffer area with a rectangle, oval or diamond shape for each feature of the input point layer.

The shape parameters can be fixed for all features or dynamic using a field or an expression.



24.134 Fig. : Different buffer shapes with dynamic parameters

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: taškas]	Input point vector layer
Buffer shape	`SHAPE`	[sąrašas]	The shape to use. One of: • 0 — Rectangles • 1 — Ovals • 2 — Diamonds
Width	`WIDTH`	[skaičius: slankaus kablelio] Numatytas: 1.0	Width of the buffer shape
Height	`HEIGHT`	[skaičius: slankaus kablelio] Numatytas: 1.0	Height of the buffer shape
Rotation Pasirinktinis	`ROTATION`	[skaičius: slankaus kablelio] Numatytas: 0.0	Rotation of the buffer shape
Number of segments	`SEGMENTS`	[skaičius: sveikas skaičius] Default: 36	Number of segments for a full circle (*Ovals* shape)
Išvestis	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvestis	`OUTPUT`	[vektorius: poligonas]	The output vector layer (with the buffer shapes)

Pythono kodas

Algorithm ID: `native:rectanglesovalsdiamonds`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Remove duplicate vertices

Removes duplicate vertices from features, wherever removing the vertices does not result in a degenerate geometry.

The tolerance parameter specifies the tolerance for coordinates when determining whether vertices are identical.

By default, Z values are not considered when detecting duplicate vertices. E.g. two vertices with the same X and Y coordinate but different Z values will still be considered duplicate and one will be removed. If the *Use Z Value* parameter is true, then the Z values are also tested and vertices with the same X and Y but different Z will be maintained.

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Pastaba

Duplicate vertices are not tested between different parts of a multipart geometry, e.g. a multipoint geometry with overlapping points will not be changed by this method.

Taip pat žiūrėkite

Extract vertices, *Extract specific vertices*, *Delete duplicate geometries*, *Delete duplicated vertices*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Tolerancija	`TOLERANCE`	[skaičius: slankaus kablelio] Default: 0.000001	Vertices closer than the specified distance are considered duplicates

continues on next page

Table 24.712 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Use Z value	`USE_Z_VALUE`	[loginis] Numatytas: ne	If the *Use Z Value* parameter is true, then the Z values are also tested and vertices with the same X and Y but different Z will be maintained.
Cleaned	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Cleaned	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer (without duplicate vertices)

Pythono kodas

Algorithm ID: `native:removeduplicatevertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Remove null geometries

Removes any features which do not have a geometry from a vector layer. All other features will be copied unchanged.

The features with null geometries can be saved to a separate layer.

If *Also remove empty geometries* is checked, the algorithm removes features whose geometries have no coordinates, i.e., geometries that are empty. In that case, also the null output will reflect this option, containing both null and empty geometries.

Taip pat žiūrėkite

Delete duplicate geometries

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Input vector layer (with non-NULL geometries)
Also remove empty geometries	`REMOVE_EMPTY`	[loginis]	
Non null geometries	`OUTPUT` Pasirinktinis	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer for the non-NULL (and non-empty) geometries. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Null geometries Pasirinktinis	`NULL_OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Praleisti išvestį]`	Specify the output vector layer for the NULL (and empty) geometries. *One of*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Null geometries	`NULL_OUTPUT`	[toks pat, kaip įvestis]	Output vector layer (for NULL and, if chosen, empty geometries)
Non null geometries	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer (without NULL and, if chosen, empty geometries)

Pythono kodas

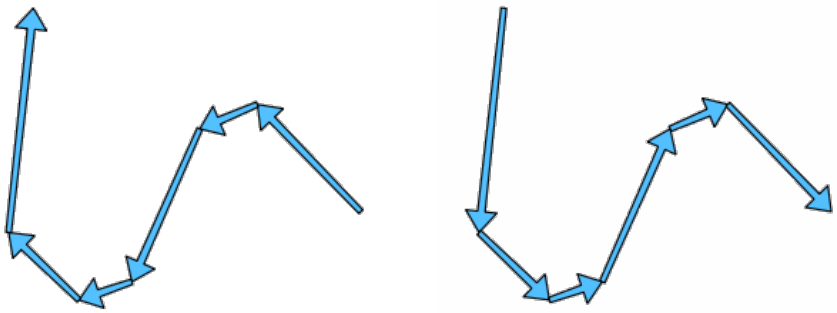
Algorithm ID: `native:removenullgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Reverse line direction

Inverts the direction of a line layer.



24.135 Fig. : Before and after the direction inversion

Leidžia linijų *geoobjektus keisti vietoje*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Reversed	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output line vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Reversed	`OUTPUT`	[vektorius: linija]	The output line vector layer (with reversed lines)

Pythono kodas

Algorithm ID: `native:reverselinedirection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Rotate

Rotates feature geometries by the specified angle clockwise. The rotation occurs around each feature's centroid, or optionally around a unique preset point.

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Taip pat žiūrėkite

Translate, *Swap X and Y coordinates*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Rotation (degrees clockwise)	`ANGLE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Angle of the rotation in degrees
Rotation anchor point (x, y) Pasirinktinis	`ANCHOR`	[koordinatės] Numatytas: jokio	X,Y coordinates of the point to rotate the features around. If not set the rotation occurs around each feature's centroid.
Rotated	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer (with rotated geometries). *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Rotated	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer with rotated geometries

Pythono kodas

Algorithm ID: `native:rotatefeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Roundness

Calculates the roundness of each feature and stores it as a new field `roundness`. The input vector layer must contain polygons.

The roundness of a polygon is defined as $4\pi \times$ polygon area / perimeter^2. The roundness value varies between 0 and 1. A perfect circle has a roundness of 1, while a completely flat polygon has a roundness of 0.

Pastaba

The algorithm returns NULL for multipart polygon features.

Leidžia *vietoje keisti* poligonų geoobjektus

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Įvesties vektorinis sluoksnis
Roundness	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer (with roundness field). *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Roundness	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer with `roundness` value in a field

Pythono kodas

Algorithm ID: `native:roundness`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Segmentize by maximum angle

Segmentizes a geometry by converting curved sections to linear sections.

The segmentization is performed by specifying the maximum allowed radius angle between vertices on the straightened geometry (e.g the angle of the arc created from the original arc center to consecutive output vertices on the linearized geometry). Non-curved geometries will be retained without change.

Taip pat žiūrėkite

Segmentize by maximum distance, *Simplify*, *Smooth*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija, poligonas]	Įvesties linijų ar poligonų vektorinis sluoksnis
Maximum angle between vertices (degrees)	`ANGLE`	[skaičius: slankaus kablelio] Default: 5.0	Maximum allowed radius angle between vertices on the straightened geometry
Segmentized	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer (with segmentized geometries). *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Segmentized	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer with segmentized geometries

Pythono kodas

Algorithm ID: `native:segmentizebymaxangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Segmentize by maximum distance

Segmentizes a geometry by converting curved sections to linear sections.

The segmentization is performed by specifying the maximum allowed offset distance between the original curve and the segmentized representation. Non-curved geometries will be retained without change.

Taip pat žiūrėkite

Segmentize by maximum angle, *Simplify*, *Smooth*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija, poligonas]	Įvesties linijų ar poligonų vektorinis sluoksnis
Maximum offset distance	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Maximum allowed offset distance between the original curve and the segmentized representation, in the layer units.
Segmentized	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer (with segmentized geometries). *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Segmentized	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer with segmentized geometries

Pythono kodas

Algorithm ID: `native:segmentizebymaxdistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Set M value

Sets the M value for geometries in a layer.

If M values already exist in the layer, they will be overwritten with the new value. If no M values exist, the geometry will be upgraded to include M values and the specified value used as the initial M value for all geometries.

Allows *features in-place modification* of point, line, and polygon features with M enabled

Patarimas

Use the Identify Features button to check the added M value: the results are available in the *Identify Results* dialog.

Taip pat žiūrėkite

Set M value from raster, *Set Z value*, *Drop M/Z values*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
M Value	`M_VALUE`	[skaičius: slankaus kablelio] Numatytas: 0.0	M value to assign to the feature geometries
M Added	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
M Added	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer (with M values assigned to the geometries)

Pythono kodas

Algorithm ID: `native:setmvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Set M value from raster

Uses values sampled from a band within a raster layer to set the M value for every overlapping vertex in the feature geometry. The raster values can optionally be scaled by a preset amount.

If M values already exist in the layer, they will be overwritten with the new value. If no M values exist, the geometry will be upgraded to include M values.

Allows *features in-place modification* of point, line, and polygon features with M enabled

Taip pat žiūrėkite

Drape (set Z value from raster), *Set M value*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Rastro sluoksnis	`RASTER`	[rastras]	Raster layer with M values
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	The raster band from which the M values are taken
Value for NoData or non-intersecting vertices	`NODATA`	[skaičius: slankaus kablelio] Numatytas: 0.0	Value to use in case the vertex does not intersect (a valid pixel of) the raster
Scale factor	`SCALE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Scaling value: the band values are multiplied by this value.
Offset	`OFFSET`	[skaičius: slankaus kablelio] Numatytas: 0.0	Offset value: it is algebraically added to the band values after applying the „Scale factor“.
Updated	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer (with updated M values). *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Updated	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer (with updated M values)

Pythono kodas

Algorithm ID: `native:setmfromraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Set Z value

Sets the Z value for geometries in a layer.

If Z values already exist in the layer, they will be overwritten with the new value. If no Z values exist, the geometry will be upgraded to include Z values and the specified value used as the initial Z value for all geometries.

Allows *features in-place modification* of point, line, and polygon features with Z enabled

Patarimas

Use the Identify Features button to check the added Z value: the results are available in the *Identify Results* dialog.

Taip pat žiūrėkite

Drape (set Z value from raster), *Set M value*, *Drop M/Z values*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Z Value	`Z_VALUE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Z value to assign to the feature geometries

continues on next page

Table 24.730 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Z Added	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Z Added	`OUTPUT`	[toks pat, kaip įvestis]	The output vector layer (with Z values assigned)

Pythono kodas

Algorithm ID: `native:setzvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Simplify

Simplifies the geometries in a line or polygon layer. It creates a new layer with the same features as the ones in the input layer, but with geometries containing a lower number of vertices.

The algorithm gives a choice of simplification methods, including distance based (the „Douglas-Peucker“ algorithm), area based („Visvalingam“ algorithm) and snapping geometries to grid.

24.136 Fig. : From left to right, source layer and increasing simplification tolerances

Allows *features in-place modification* of line and polygon features

Numatytas meniu: *Vektorius ► Geometrijos įrankiai*

Taip pat žiūrėkite

Paprastinti padengimą, *Smooth*, *Densify by count*, *Densify by interval*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija, poligonas]	Įvesties linijų ar poligonų vektorinis sluoksnis
Simplification method	`METHOD`	[sąrašas] Numatytas: 0	Simplification method. One of: • 0 — Distance (Douglas-Peucker) • 1 — Snap to grid • 2 — Area (Visvalingam)
Tolerancija	`TOLERANCE`	[skaičius: slankaus kablelio] Numatytas: 1.0	Threshold tolerance (in units of the layer): if the distance between two nodes is smaller than the tolerance value, the segment will be simplified and vertices will be removed.
Supaprastinta	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output (simplified) vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Supaprastinta	`OUTPUT`	[toks pat, kaip įvestis]	The output (simplified) vector layer

Pythono kodas

Algorithm ID: `native:simplifygeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Single sided buffer

Computes a buffer on lines by a specified distance on one side of the line only.

Buffer always results in a polygon layer.

24.137 Fig. : Left versus right side buffer on the same vector line layer

Taip pat žiūrėkite

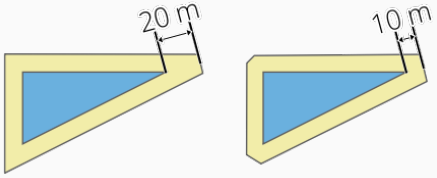
Buferis

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Atstumas	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 10.0	Buffer distance.
Side	`SIDE`	[sąrašas] Numatytas: 0	Which side to create the buffer on. One of: • 0 – Left • 1 – Right
Segmentai	`SEGMENTS`	[skaičius: sveikas skaičius] Numatyta: 8	Valdo skaičių linijų segmentų, kurie naudojami aproksimuojant apskritimo ketvirčius kuriant užapvalintus poslinkius.
Jungimo stilius	`JOIN_STYLE`	[sąrašas] Numatytas: 0	Nurodo, ar paslinkus linijos kampus reikia naudoti apvalias, mitros ar kampines jungtis. Parinktys yra: • 0 — Apvalūs • 1 — Mitra • 2 — Kampas 24.138 Fig. : Apvalūs, mitros ir kampo jungimo stiliai

continues on next page

Table 24.734 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Mitros riba	`MITER_LIMIT`	[skaičius: slankaus kablelio] Numatyta: 2.0	Nustato maksimalų atstumą nuo paslenkamos geometrijos kuriant mitros jungtį. Nurodomas kaip poslinkio atstumo koeficientas (taikoma tik mitros jungimo stiliui). Minimalus: 1.0  24.139 Fig. : 10m buferis su 2 riba ir 10m buferis su 1 riba
Buferis	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties (buferių) sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Buferis	`OUTPUT`	[vektorius: poligonas]	Išvesties (buferių) poligonų sluoksnis

Pythono kodas

Algorithm ID: `native:singlesidedbuffer`

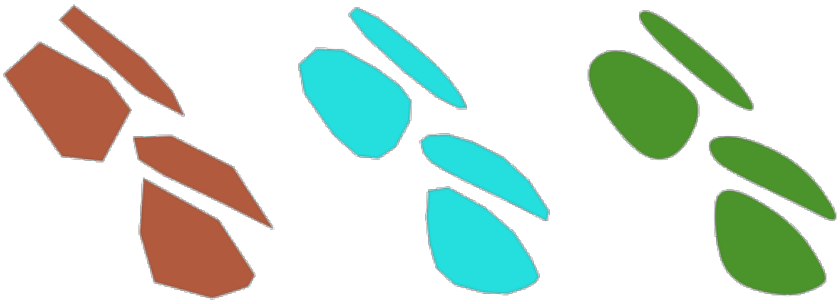
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Smooth

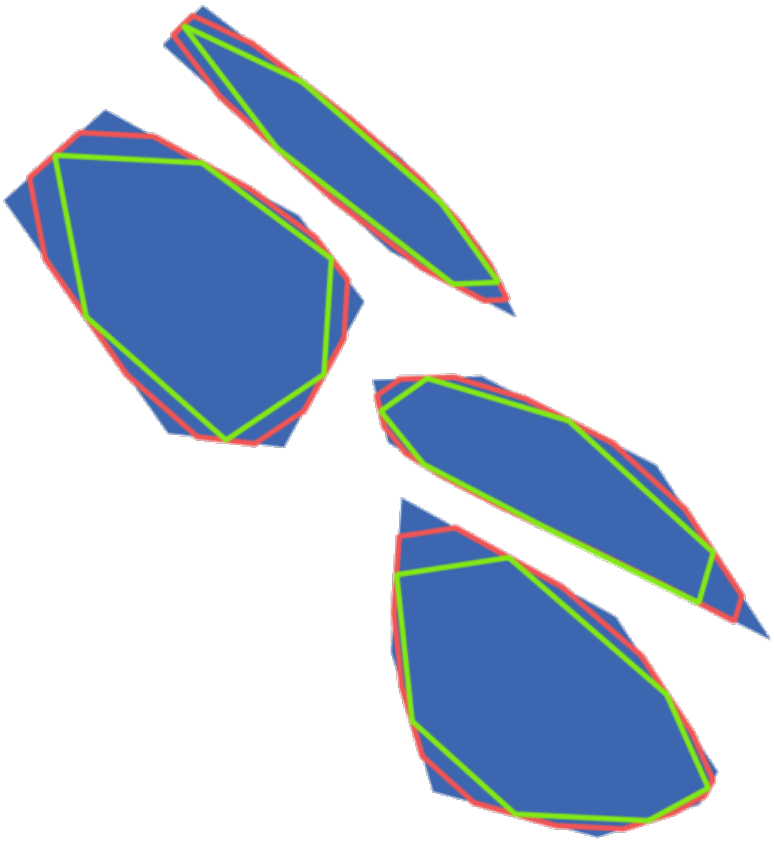
Smooths the geometries in a line or polygon layer by adding more **vertices and corners** to the feature geometries.

The iterations parameter dictates how many smoothing iterations will be applied to each geometry. A higher number of iterations results in smoother geometries with the cost of greater number of nodes in the geometries.



24.140 Fig. : Increasing number of iterations causes smoother geometries

The offset parameter controls how „tightly“ the smoothed geometries follow the original geometries. Smaller values results in a tighter fit, and larger values will create a looser fit.



24.141 Fig. : Blue: the input layer. Offset 0.25 gives the red line, while offset 0.50 gives the green line.

The maximum angle parameter can be used to prevent smoothing of nodes with large angles. Any node where the angle of the segments to either side is larger than this will not be smoothed. For example, setting the maximum angle to 90 degrees or lower would preserve right angles in the geometry.

Allows *features in-place modification* of line and polygon features

Taip pat žiūrėkite

Simplify, *Paprastinti padengimą*, *Densify by count*, *Densify by interval*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija, poligonas]	Įvesties linijų ar poligonų vektorinis sluoksnis
Iterations	`ITERATIONS`	[skaičius: sveikas skaičius] Numatytas: 1	Increasing the number of iterations will give smoother geometries (and more vertices).
Offset	`OFFSET`	[skaičius: slankaus kablelio] Numatyta: 0.25	Increasing values will *move* the smoothed lines / boundaries further away from the input lines / boundaries.
Maximum node angle to smooth	`MAX_ANGLE`	[skaičius: slankaus kablelio] Default: 180.0	Every node below this value will be smoothed
Smoothed	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output (smoothed) layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Smoothed	`OUTPUT`	[toks pat, kaip įvestis]	Output (smoothed) vector layer

Pythono kodas

Algorithm ID: `native:smoothgeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Snap geometries to layer

Snaps the geometries in a layer either to the geometries from another layer, or to geometries within the same layer.

Matching is done based on a tolerance distance, and vertices will be inserted or removed as required to make the geometries match the reference geometries.

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Taip pat žiūrėkite

Snap points to grid

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Atskaitos sluoksnis	`REFERENCE_LAYE`	[vektorius: geometrija]	Vector layer to snap to
Tolerancija	`TOLERANCE`	[skaičius: slankaus kablelio] Numatytas: 10.0	Control how close input vertices need to be to the reference layer geometries before they are snapped.

continues on next page

Table 24.738 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Behavior	`BEHAVIOR`	[sąrašas] Numatytas: 0	Snapping can be done to an existing node or a segment (its closest point to the vertex to move). Available snapping options: • 0 — Prefer aligning nodes, insert extra vertices where required Prefer to snap to nodes, even when a segment may be closer than a node. New nodes will be inserted to make geometries follow each other exactly when inside allowable tolerance. • 1 — Prefer closest point, insert extra vertices where required Snap to closest point, regardless of it is a node or a segment. New nodes will be inserted to make geometries follow each other exactly when inside allowable tolerance. • 2 — Prefer aligning nodes, don't insert new vertices Prefer to snap to nodes, even when a segment may be closer than a node. No new nodes will be inserted. • 3 — Prefer closest point, don't insert new vertices Snap to closest point, regardless of it is a node or a segment. No new nodes will be inserted. • 4 — Move end points only, prefer aligning nodes Only snap start/end points of lines (point features will also be snapped, polygon features will not be modified), prefer to snap to nodes. • 5 — Move end points only, prefer closest point Only snap start/end points of lines (point features will also be snapped, polygon features will not be modified), snap to closest point • 6 — Snap end points to end points only Only snap the start/end points of lines to other start/end points of lines • 7 — Snap to anchor nodes (single layer only)
Snapped geometry	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output (snapped) layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Snapped geometry	`OUTPUT`	[toks pat, kaip įvestis]	Output (snapped) vector layer

Pythono kodas

Algorithm ID: `native:snapgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Snap points to grid

Modifies the coordinates of geometries in a vector layer, so that all points or vertices are snapped to the closest point of a grid.

If the snapped geometry cannot be calculated (or is totally collapsed) the feature's geometry will be cleared.

Snapping can be performed on the X, Y, Z or M axis. A grid spacing of 0 for any axis will disable snapping for that axis.

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Įspėjimas

Snapping to grid may generate an invalid geometry in some corner cases.

Taip pat žiūrėkite

Snap geometries to layer

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
X Grid Spacing	`HSPACING`	[skaičius: slankaus kablelio] Numatytas: 1.0	Grid spacing on the X axis
Y Grid Spacing	`VSPACING`	[skaičius: slankaus kablelio] Numatytas: 1.0	Grid spacing on the Y axis

continues on next page

Table 24.740 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Z Grid Spacing	`ZSPACING`	[skaičius: slankaus kablelio] Numatytas: 0.0	Grid spacing on the Z axis
M Grid Spacing	`MSPACING`	[skaičius: slankaus kablelio] Numatytas: 0.0	Grid spacing on the M axis
Snapped	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output (snapped) layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Snapped	`OUTPUT`	[toks pat, kaip įvestis]	Output (snapped) vector layer

Pythono kodas

Algorithm ID: `native:snappointstogrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Split lines by maximum length

Takes a line (or curve) layer and splits each feature into multiple parts, where each part is of a specified maximum length. Z and M values at the start and end of the new line substrings are linearly interpolated from existing values.

Leidžia linijų *geoobjektus keisti vietoje*

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	The input line vector layer
Maximum line length	`LENGTH`	[skaičius: slankaus kablelio] Numatytas: 10.0	The maximum length of a line in the output.
Split	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output line vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Split	`OUTPUT`	[vektorius: linija]	The new line vector layer. The length of the feature geometries is less than or equal to the length specified in the LENGTH parameter. An additional field containing the `order` will be added.

Pythono kodas

Algorithm ID: `native:splitlinesbylength`

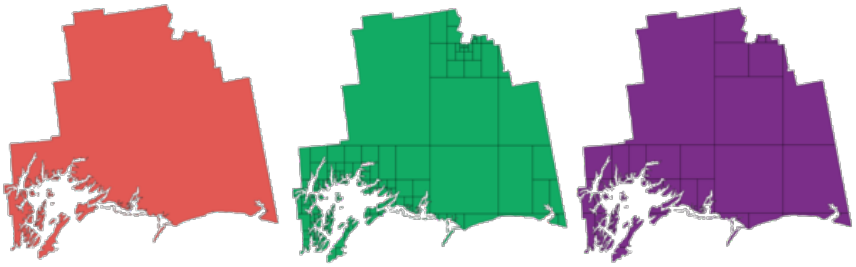
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Subdivide

Subdivides the geometry. The returned geometry will be a collection containing subdivided parts from the original geometry, where no part has more than the specified maximum number of nodes.

This is useful for dividing a complex geometry into less complex parts, easier to spatially index and faster to perform spatial operations. Curved geometries will be segmentized before subdivision.



24.142 Fig. : Left the input layer, middle maximum nodes value is 100 and right maximum value is 200

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Pastaba

Subdividing a geometry can generate geometry parts that may not be valid and may contain self-intersections.

Taip pat žiūrėkite

Explode lines, *Line substring*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Maximum nodes in parts	`MAX_NODES`	[skaičius: sveikas skaičius] Default: 256	Maximum number of vertices each new geometry part is allowed to have. Fewer *sub-parts* for higher values.
Subdivided	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output (subdivided) vector layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Subdivided	`OUTPUT`	[toks pat, kaip įvestis]	Output vector layer

Pythono kodas

Algorithm ID: `native:subdivide`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Swap X and Y coordinates

Switches the X and Y coordinate values in input geometries.

It can be used to repair geometries which have accidentally had their latitude and longitude values reversed.

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Taip pat žiūrėkite

Translate, *Rotate*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Swapped	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Swapped	`OUTPUT`	[toks pat, kaip įvestis]	Output (swapped) vector layer

Pythono kodas

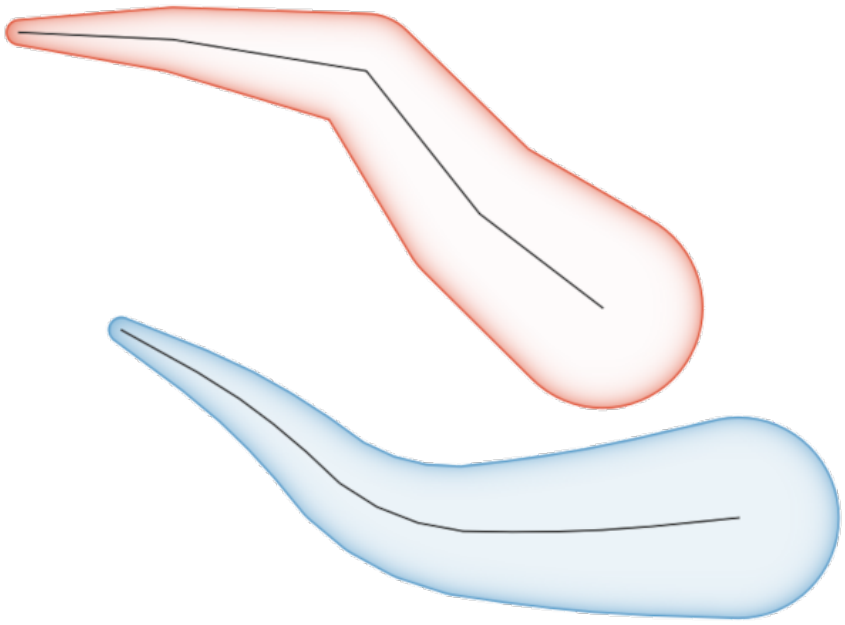
Algorithm ID: `native:swapxy`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Tapered buffers

Creates tapered buffer along line geometries, using a specified start and end buffer diameter.



24.143 Fig. : Tapered buffer example

Taip pat žiūrėkite

Variable width buffer (by M value), *Buferis*, *Create wedge buffers*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Start width	`START_WIDTH`	[skaičius: slankaus kablelio] Numatytas: 0.0	Represents the radius of the buffer applied at the start point of the line feature
End width	`END_WIDTH`	[skaičius: slankaus kablelio] Numatytas: 0.0	Represents the radius of the buffer applied at the end point of the line feature.
Segmentai	`SEGMENTS`	[skaičius: sveikas skaičius] Default: 16	Valdo skaičių linijų segmentų, kurie naudojami aproksimuojant apskritimo ketvirčius kuriant užapvalintus poslinkius.
Buferiai	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties (buferių) sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Buferiai	`OUTPUT`	[vektorius: poligonas]	Išvesties (buferių) poligonų sluoksnis

Pythono kodas

Algorithm ID: `native:taperedbuffer`

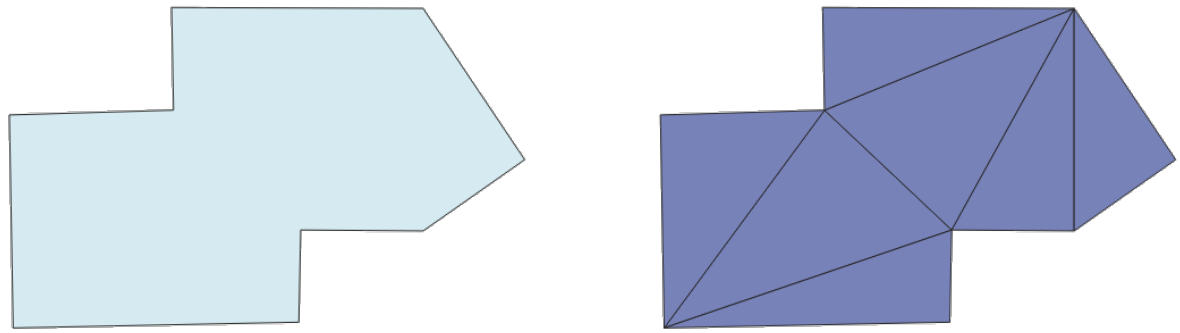
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Tessellate

Tessellates a polygon geometry layer, dividing the geometries into triangular components.

The output layer consists of multipolygon geometries for each input feature, with each multipolygon consisting of multiple triangle component polygons.



24.144 Fig. : Tessellated polygon (right)

Leidžia *vietoje keisti* poligonų geoobjektus

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: poligonas]	Įvesties poligonų vektorinis sluoksnis
Tesselated	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Tesselated	`OUTPUT`	[vektorius: poligonas]	Output multipolygonZ layer

Pythono kodas

Algorithm ID: `3d:tessellate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Transect

Creates transects on vertices for (multi)linestring.

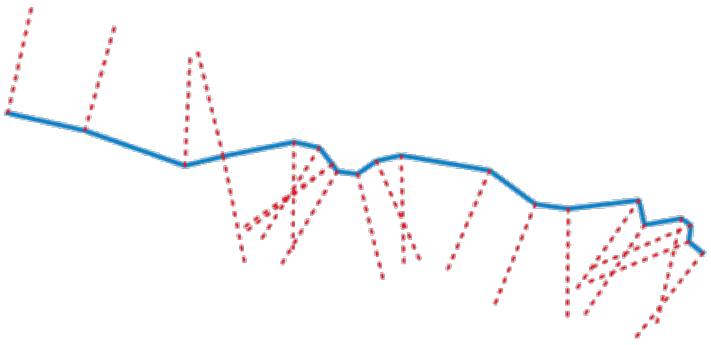
A transect is a line oriented from an angle (by default perpendicular) to the input polylines (at vertices).

Field(s) from feature(s) are returned in the transect with these new fields:

- TR_FID: ID of the original feature
- TR_ID: ID of the transect. Each transect have an unique ID
- TR_SEGMENT: ID of the segment of the linestring
- TR_ANGLE: Angle in degrees from the original line at the vertex
- TR_LENGTH: Total length of the transect returned
- TR_ORIENT: Side of the transect (only on the left or right of the line, or both side)

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.



24.145 Fig. : Dashed red lines represent the transect of the input line layer

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Length of the transect	`LENGTH`	[skaičius: slankaus kablelio] Default: 5.0	Length in map unit of the transect
Angle in degrees from the original line at the vertices	`ANGLE`	[skaičius: slankaus kablelio] Default: 90.0	Change the angle of the transect
Side to create the transect	`SIDE`	[sąrašas]	Choose the side of the transect. Available options are: • 0 — Left • 1 — Right • 2 — Both
Transect	`OUTPUT`	[vektorius: linija] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output line layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Transect	`OUTPUT`	[vektorius: linija]	Output line layer

Pythono kodas

Algorithm ID: `native:transect`

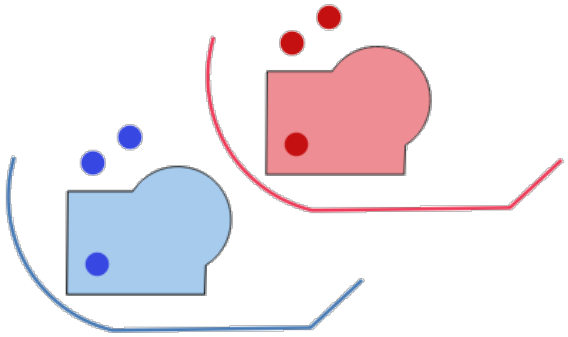
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Translate

Moves the geometries within a layer, by offsetting with a predefined X and Y displacement.

Z and M values present in the geometry can also be translated.



24.146 Fig. : Features in red represent the translated geometry of the input (blue) layer

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Taip pat žiūrėkite

Pakeistų geoobjektų masyvas, *Offset lines*, *Rotate*, *Swap X and Y coordinates*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Offset distance (x-axis)	`DELTA_X`	[skaičius: slankaus kablelio] Numatytas: 0.0	X ašyje taikomas poslinkis
Offset distance (y-axis)	`DELTA_Y`	[skaičius: slankaus kablelio] Numatytas: 0.0	Y ašyje taikomas poslinkis
Offset distance (z-axis)	`DELTA_Z`	[skaičius: slankaus kablelio] Numatytas: 0.0	Z ašyje taikomas poslinkis
Offset distance (m values)	`DELTA_M`	[skaičius: slankaus kablelio] Numatytas: 0.0	Displacement to apply on the M axis

continues on next page

Table 24.754 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Pakeista	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Pakeista	`OUTPUT`	[toks pat, kaip įvestis]	Output vector layer

Pythono kodas

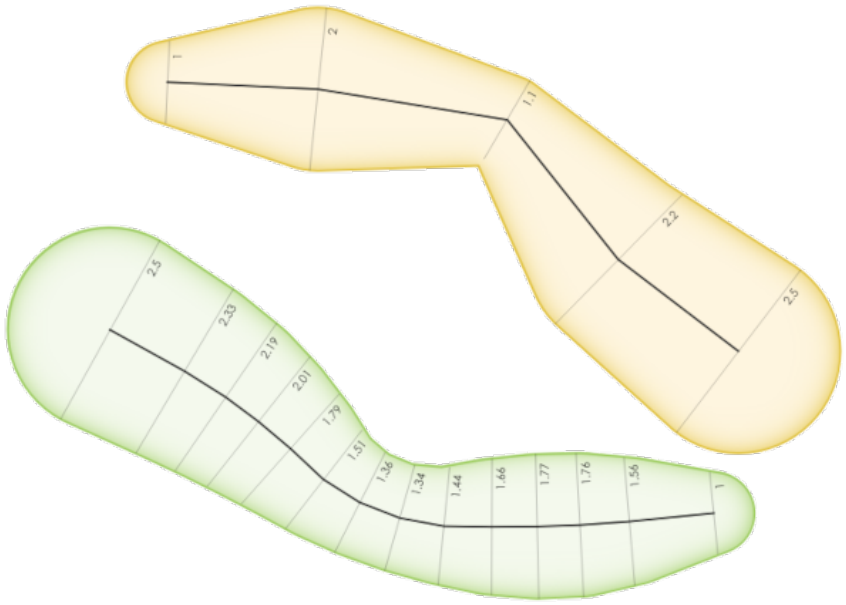
Algorithm ID: `native:translategeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Variable width buffer (by M value)

Creates variable width buffers along lines, using the M value of the line geometries as the diameter of the buffer at each vertex.



24.147 Fig. : Variable buffer example

Taip pat žiūrėkite

Tapered buffers, *Buferis*, *Set M value*, *Variable distance buffer*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Įvesties vektorinis linijų sluoksnis
Segmentai	`SEGMENTS`	[skaičius: sveikas skaičius] Default: 16	Number of the buffer segments per quarter circle. It can be a unique value (same value for all the features), or it can be taken from features data (the value can depend on feature attributes).
Buferiai	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties (buferių) sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Buferiai	`OUTPUT`	[vektorius: poligonas]	Variable buffer polygon layer

Pythono kodas

Algorithm ID: `native:bufferbym`

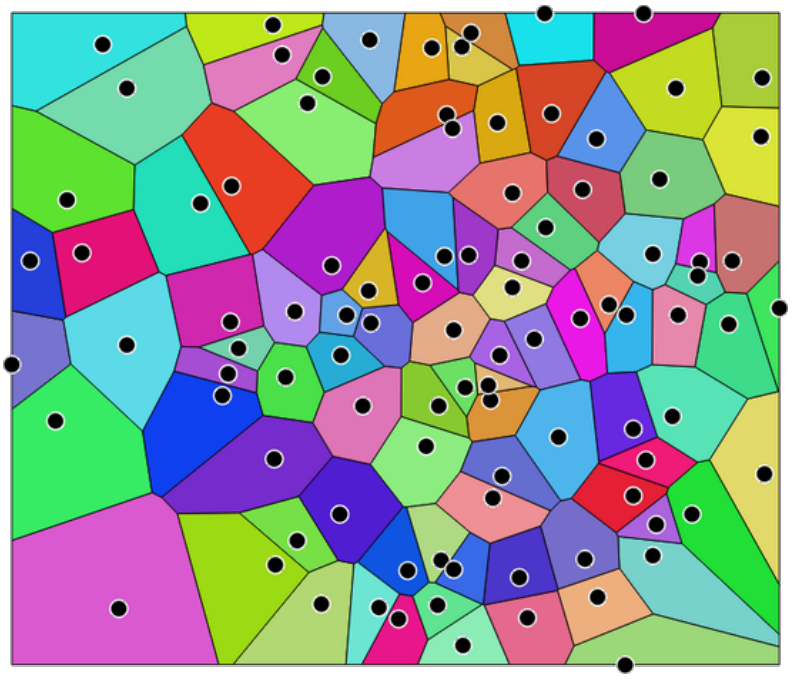
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Voronoi polygons

Takes a point layer and generates a polygon layer containing the Voronoi polygons (known also as Thiessen polygons) corresponding to those input points.

Any location within a Voronoi polygon is closer to the associated point than to any other point.



24.148 Fig. : Voronoi polygons

Numatytas meniu: *Vektorius* ► *Geometrijos įrankiai*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: taškas]	Input point vector layer
Buffer region (% of extent)	`BUFFER`	[skaičius: slankaus kablelio] Numatytas: 0.0	The extent of the output layer will be this much bigger than the extent of the input layer
Tolerancija Pasirinktinis	`TOLERANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Specifies an optional snapping tolerance which can be used to improve the robustness of the voronoi.
Copy attributes from input features	`COPY_ATTRIBUTE`	[loginis] Numatytas: Taip	Specifies whether fields storing involved point features ID should be added to the output. If False, an `id` field is created to identify the polygons.
Voronoi polygons	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output layer (with the Voronoi polygons). *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Voronoi polygons	`OUTPUT`	[vektorius: poligonas]	Voronoi polygons of the input point vector layer

Pythono kodas

Algorithm ID: `native:voronoipolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.27 Vektorių persidengimai

Apkirpti

Apkerpa vektorinį sluoksnį naudojant geoobjektus iš papildomo poligonų sluoksnio.

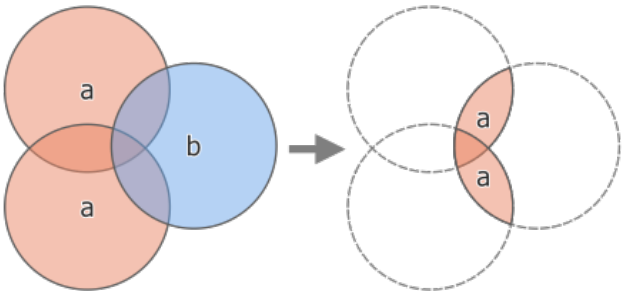
Į rezultatų sluoksnį bus įtrauktos tik tos pradinio sluoksnio geoobjektų dalys, kurios patenka į persidengimo sluoksnio poligonus.

Įspėjimas

Keičiama tik geometrija

Šis veiksmas keičia tik geoobjekto geometriją. Geoobjektų atributų reikšmės **nekeičiamos**, nors geoobjektų savybės, tokios kaip plotas ar ilgis, gali pasikeisti. Jei šios savybės laikomos atributuose, juos reikės pakeisti rankiniu būdu.

Šis algoritmas naudoja tiekėjų erdvinius indeksus, paruoštas geometrijas ir apkarpymo veiksmą taiko tik jei geometrija nepatenka pilnai į slėpimo geometriją.



24.149 Fig. : Karpymo veiksmas su dviejų geoobjektų įvesties sluoksniu „a“ ir vieno geoobjekto perdengimo sluoksniu „b“ (kairėje) - gaunamas naujas sluoksnis su pakeistais „a“ geoobjektais (dešinėje)

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Numatytas meniu: *Vektorius ► Geometrijos įrankiai*

Taip pat žiūrėkite

Intersection, *Skirtumas*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Sluoksnis su apkerpamais geoobjektais
Perdengimo sluoksnis	`OVERLAY`	[vektorius: poligonas]	Sluoksnis su apkirpimo geoobjektais

continues on next page

Table 24.760 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Apkirpta	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite sluoksnį, kuriame turės būti įvesties sluoksnio geoobjektai, kurie yra perdangos (apkarpymo) sluoksnio viduje. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Apkirpta	`OUTPUT`	[toks pat, kaip įvestis]	Sluoksnis su geoobjektais iš įvesties sluoksnio, padalintais perdengimo sluoksniu.

Pythono kodas

Algorithm ID: `native:clip`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Skirtumas

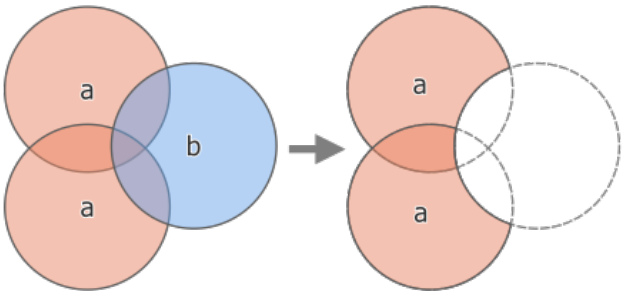
Ištraukia iš įvesties sluoksnio geoobjektus, kurie nepatenka į persidengimo sluoksnio ribas.

Įvesties sluoksnio geoobjektai, kurie dalinai kertasi su persidengimo sluoksnio geoobjektu(ais) padalinami pagal tų objektų ribą ir paliekamos tik tos dalys, kurios yra už persidengimo sluoksnio objektų.

Įspėjimas

Keičiama tik geometrija

Šis veiksmas keičia tik geoobjekto geometriją. Geoobjektų atributų reikšmės **nekeičiamos**, nors geoobjektų savybės, tokios kaip plotas ar ilgis, gali pasikeisti. Jei šios savybės laikomos atributuose, juos reikės pakeisti rankiniu būdu.



24.150 Fig. : Skirtumo veiksmas su dviem geoobjektais įvesties sluoksnyje „a“ ir vienu geoobjektu persidengimo sluoksnyje „b“ (kairėje) - gaunamas naujas sluoksnis su modifikuotais „a“ geoobjektais (dešinėje)

Leidžia taškų, linijų ir poligonų *geoobjektus keisti vietoje*

Numatytas meniu: *Vektorius ► Geometrijos įrankiai*

Taip pat žiūrėkite

Skirtumas (keli), *Symmetrical difference*, *Apkirpti*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Sluoksnis, iš kurio reikia ištraukti geoobjektus (ar jų dalis).
Perdengimo sluoksnis	`OVERLAY`	[vektorius: geometrija]	Sluoksnis su geometrijomis, kurios bus atimamos iš įvesties sluoksnio geometrijų. Tikimasi, kad jame bus bent tiek matmenų (taškai: 0D, linijos: 1D, poligonai: 2D, tūris: 3D), kiek įvesties sluoksnio geometrijose.
Skirtumas	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite sluoksnį, kuriame turės būti įvesties sluoksnio geoobjektai (jų dalys), kurie nėra perdangos (apkarpymo) sluoksnio viduje. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Tinklelio dydis Pasirinktinis	`GRID_SIZE`	[skaičius: slankaus kablelio] Numatytas: jokio	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Skirtumas	`OUTPUT`	[toks pat, kaip įvestis]	Sluoksnis su geoobjektais (ar jų dalimis) iš įvesties sluoksnio, nepersidengiančiais su persidengimo sluoksniu.

Pythono kodas

Algorithm ID: `native:difference`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Skirtumas (keli)

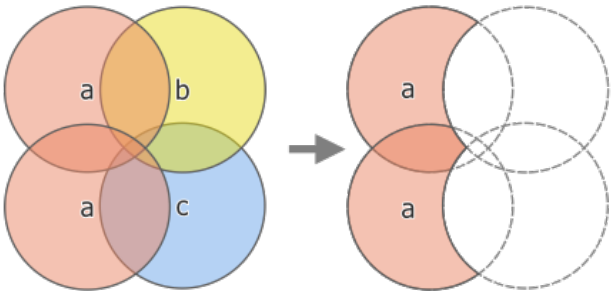
Ištraukia išvesties sluoksnio geoobjektus, kurie yra pilnai už arba dalinai dengia persidengimo sluoksnio(ių) geoobjektus.

Skirtumas kiekvienam persidengimo sluoksniui skaičiuojamas tarp visų ankstesnių skirtumo veiksmų rezultato ir perdangos sluoksnio. Įvesties sluoksnių geoobjektai, kurie dalinai dengia geoobjektą(us) perdengimo sluoksnyje, padalinami ties tų geoobjektų riba ir tik paliekamos tik tos jų dalys, kurios yra už perdengimo sluoksnio geoobjektų.

Įspėjimas

Keičiama tik geometrija

Šis veiksmas keičia tik geoobjekto geometriją. Geoobjektų atributų reikšmės **nekeičiamos**, nors geoobjektų savybės, tokios kaip plotas ar ilgis, gali pasikeisti. Jei šios savybės laikomos atributuose, juos reikės pakeisti rankiniu būdu.



24.151 Fig. : Skirtumo veiksmas tarp dviejų geoobjektų įvesties sluoksnio „a“ ir vieno geoobjekto perdengimo sluoksnių „b“ ir „c“ (kairėje) - gauname nauje sluoksnį su pakeistais „a“ geoobjektais (dešinėje)

Taip pat žiūrėkite

Skirtumas, *Symmetrical difference*, *Apkirpti*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Sluoksnis, iš kurio reikia ištraukti geoobjektus (ar jų dalis).
Perdengimo sluoksniai	`OVERLAYS`	[vektorius: geometrija] [sąrašas]	Sąrašas sluoksnų su geometrijomis, kurios bus atimamos iš įvesties sluoksnio geometrijų. Tikimasi, kad juose bus bent tiek matmenų (taškai: 0D, linijos: 1D, poligonai: 2D, tūris: 3D), kiek įvesties sluoksnio geometrijose.
Skirtumas	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite sluoksnį, kuriame bus įvesties sluoksnio geoobjektai (ar jų dalys), kurie nesidengia su persidengimo sluoksnių geoobjektais. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Skirtumas	`OUTPUT`	[toks pat, kaip įvestis]	Sluoksnis su įvesties sluoksnio geoobjektais (ar jų dalimis), kurie nesidengia su persidengimo sluoksnio geoobjektais.

Pythono kodas

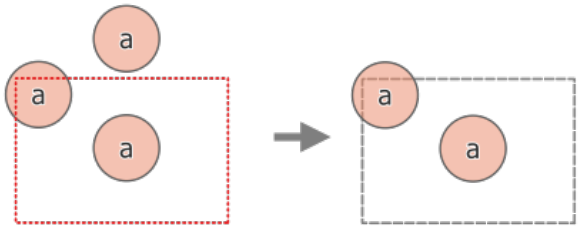
Algorithm ID: `native:multidifference`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

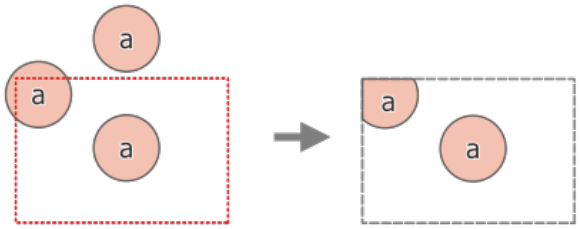
Extract/clip by extent

Creates a new vector layer that only contains features which fall within a specified extent. Any features which intersect the extent will be included by default.



24.152 Fig. : Extract operation between a three-feature input layer ‚a' and a dashed extent (left) - resulting features with dashed extent for reference (right)

Optionally, feature geometries can also be clipped to the extent.



24.153 Fig. : Extract operation between a three-feature input layer ‚a' and a dashed extent (left) - resulting features with dashed extent for reference (right)

Taip pat žiūrėkite

Apkirpti

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Sluoksnis, iš kurio reikia ištraukti geoobjektus (ar jų dalis).
Apimtis (xmin, xmax, ymin, ymax)	`EXTENT`	[apimtis]	Extent for clipping. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Clip features to extent	`CLIP`	[boolean] Numatytas: ne	If checked, the geometries will be clipped to the extent chosen instead of taking the whole geometry as output. Moreover, output geometries will be automatically converted to multi geometries to ensure uniform output types.
Extracted	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the layer to contain the features from the input layer that are inside the clip extent. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Extracted	`OUTPUT`	[toks pat, kaip įvestis]	Layer containing the clipped features.

Pythono kodas

Algorithm ID: `native:extractbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Intersection

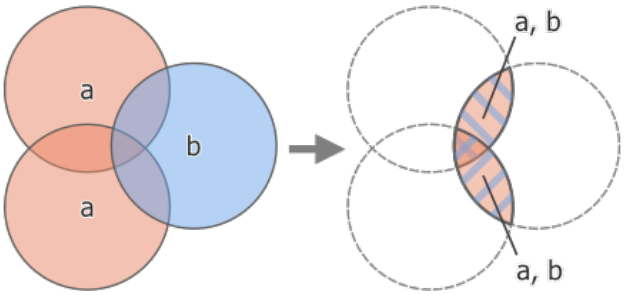
Extracts the portions of features from the input layer that overlap features in the overlay layer.

Features in the intersection layer are assigned the attributes of the overlapping features from both the input and overlay layers.

Įspėjimas

Keičiama tik geometrija

Šis veiksmas keičia tik geoobjekto geometriją. Geoobjektų atributų reikšmės **nekeičiamos**, nors geoobjektų savybės, tokios kaip plotas ar ilgis, gali pasikeisti. Jei šios savybės laikomos atributuose, juos reikės pakeisti rankiniu būdu.



24.154 Fig. : Intersection operation between a two-feature input layer ‚a' and a single feature overlay layer ‚b' (left) - overlapping areas become a new two-feature layer with both layers' attributes (right)

Numatytas meniu: *Vektorius* ► *Geometrijos įrankiai*

Įspėjimas

Šis algoritmas išmeta esamus pirminius raktus ar FID reikšmes ir perkuria jas išvesties sluoksniuose.

Taip pat žiūrėkite

Intersection (multiple), *Apkirpti*, *Skirtumas*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Sluoksnis, iš kurio reikia ištraukti geoobjektus (ar jų dalis).
Perdengimo sluoksnis	`OVERLAY`	[vektorius: geometrija]	Layer containing the features to check for overlap. Its features' geometry is expected to have at least as many dimensions (point: 0D, line: 1D, polygon: 2D, volume: 3D) as the input layer's.
Input fields to keep (leave empty to keep all fields) Pasirinktinis	`INPUT_FIELDS`	[lentelės laukas: bet koks] [sąrašas] Numatytas: jokio	Field(s) of the input layer to keep in the output. If no fields are chosen all fields are taken.
Overlay fields to keep (leave empty to keep all fields) Pasirinktinis	`OVERLAY_FIELDS`	[lentelės laukas: bet koks] [sąrašas] Numatytas: jokio	Field(s) of the overlay layer to keep in the output. If no fields are chosen all fields are taken. Duplicate field names will be appended a count suffix to avoid collision.
Intersection	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the layer to contain (the parts of) the features from the input layer that overlap one or more features from the overlay layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Overlay fields prefix Pasirinktinis	`OVERLAY_FIELDS_`	[tekstas]	Add a prefix to identify fields of the overlay layer. Duplicate field names will be appended a count suffix to avoid collision.
Tinklelio dydis Pasirinktinis	`GRID_SIZE`	[skaičius: slankaus kablelio] Numatytas: jokio	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Intersection	`OUTPUT`	[toks pat, kaip įvestis]	Layer containing (parts of) features from the input layer that overlap the overlay layer.

Pythono kodas

Algorithm ID: `native:intersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Intersection (multiple)

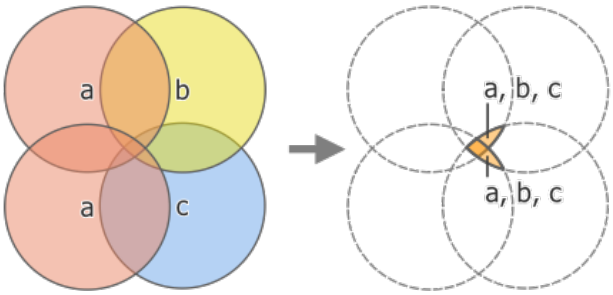
Extracts the overlapping portions of features in the input and all overlay layers.

Features in the output layer are assigned the attributes of the overlapping features from both the input and overlay layers.

Įspėjimas

Keičiama tik geometrija

Šis veiksmas keičia tik geoobjekto geometriją. Geoobjektų atributų reikšmės **nekeičiamos**, nors geoobjektų savybės, tokios kaip plotas ar ilgis, gali pasikeisti. Jei šios savybės laikomos atributuose, juos reikės pakeisti rankiniu būdu.



24.155 Fig. : Intersection operation between a two-feature input layer ‚a' and single feature overlay layers ‚b' and ‚c' (left) - overlapping areas become a new two-feature layer with all layers' attributes (right)

Įspėjimas

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Taip pat žiūrėkite

Intersection, *Apkirpti*, *Skirtumas*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Sluoksnis, iš kurio reikia ištraukti geoobjektus (ar jų dalis).
Perdengimo sluoksniai	`OVERLAYS`	[vektorius: geometrija] [sąrašas]	Layers containing the features to check for overlap. The features' geometry is expected to have at least as many dimensions (point: 0D, line: 1D, polygon: 2D, volume: 3D) as the input layer's.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Overlay fields prefix Pasirinktinis	`OVERLAY_FIELDS_`	[tekstas]	Add a prefix to identify fields of the overlay layers. Duplicate field names will be appended a count suffix to avoid collision.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Intersection	`OUTPUT`	[toks pat, kaip įvestis]	Layer containing (parts of) features from the input layer that overlap all the overlay layers.

Pythono kodas

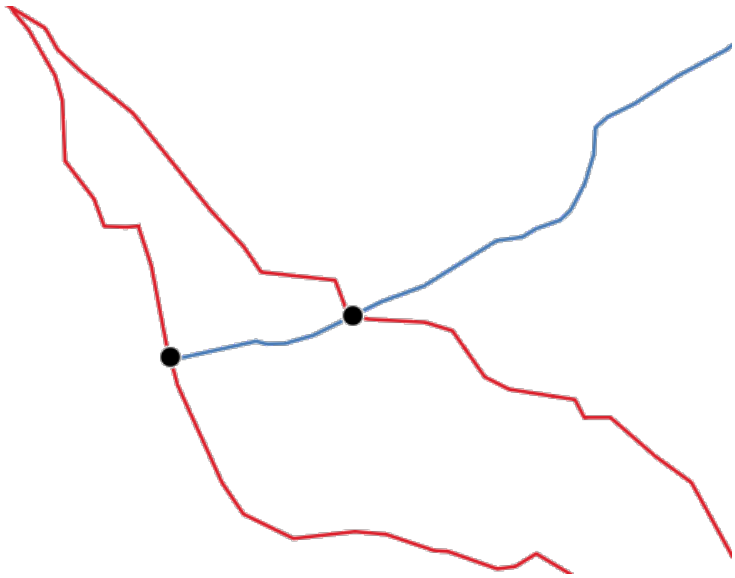
Algorithm ID: `native:multiintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Line intersections

Creates point features where the lines from the two layers intersect.



24.156 Fig. : Points of intersection

Numatytas meniu: *Vektorius* ► *Analizės įrankiai*

Įspėjimas

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	Input line layer.
Intersect layer	`INTERSECT`	[vektorius: linija]	Layer to use to find line intersections.
Input fields to keep (leave empty to keep all fields) Pasirinktinis	`INPUT_FIELDS`	[lentelės laukas: bet koks] [sąrašas] Numatytas: jokio	Field(s) of the input layer to keep in the output. If no fields are chosen all fields are taken.
Intersect fields to keep (leave empty to keep all fields) Pasirinktinis	`INTERSECT_FIEL`	[lentelės laukas: bet koks] [sąrašas] Numatytas: jokio	Field(s) of the intersect layer to keep in the output. If no fields are chosen all fields are taken. Duplicate field names will be appended a count suffix to avoid collision.

continues on next page

Table 24.775 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Intersection	`OUTPUT`	[vector: point] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the layer to contain the intersection points of the lines from the input and overlay layers. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Intersect fields prefix Pasirinktinis	`INTERSECT_FIEL`	[tekstas]	Add a prefix to identify fields of the intersect layer.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Intersections	`OUTPUT`	[vector: point]	Point vector layer of the lines intersections, with both layers' attributes.

Pythono kodas

Algorithm ID: `native:lineintersections`

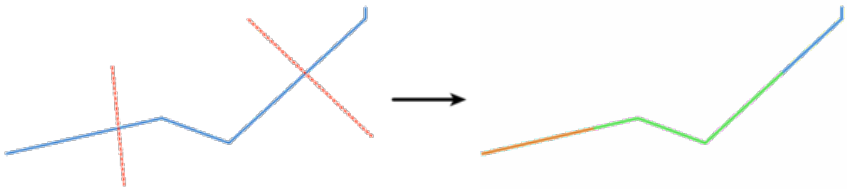
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Split with lines

Splits the lines or polygons in one layer using the lines or polygon rings in another layer to define the breaking points. Intersection between geometries in both layers are considered as split points.

Output will contain multi geometries for split features.



24.157 Fig. : Split lines

Allows *features in-place modification* of line and polygon features

Įspėjimas

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija, poligonas]	Layer containing the lines or polygons to split.
Split layer	`LINES`	[vektorius: linija, poligonas]	Layer whose lines or rings are used to define the breaking points.
Split	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the layer to contain the splitted (in case they are intersected by a line in the split layer) line/polygon features from the input layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Split	`OUTPUT`	[toks pat, kaip įvestis]	Output vector layer with split lines or polygons from input layer.

Pythono kodas

Algorithm ID: `native:splitwithlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Symmetrical difference

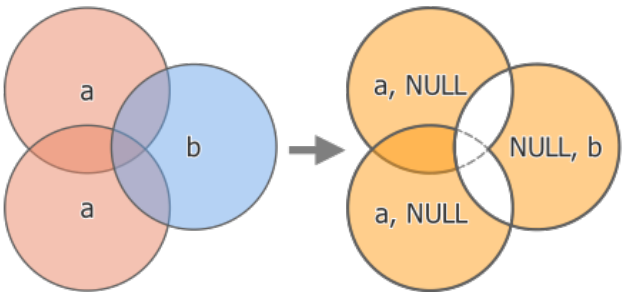
Creates a layer containing features from both the input and overlay layers but with the overlapping areas between the two layers removed.

The attribute table of the symmetrical difference layer contains attributes and fields from both the input and overlay layers.

Įspėjimas

Keičiama tik geometrija

Šis veiksmas keičia tik geoobjekto geometriją. Geoobjektų atributų reikšmės **nekeičiamos**, nors geoobjektų savybės, tokios kaip plotas ar ilgis, gali pasikeisti. Jei šios savybės laikomos atributuose, juos reikės pakeisti rankiniu būdu.



24.158 Fig. : Symmetrical difference operation between a two-features input layer ‚a' and a single feature overlay layer ‚b' (left) - resulting three-feature layer with both layers' attributes (right)

Numatytas meniu: *Vektorius* ► *Geometrijos įrankiai*

Įspėjimas

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Taip pat žiūrėkite

Skirtumas, *Apkirpti*, *Intersection*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	First layer to extract (parts of) features from.
Perdengimo sluoksnis	`OVERLAY`	[vektorius: geometrija]	Second layer to extract (parts of) features from. Ideally the geometry type should be the same as input layer.
Symmetrical difference	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the layer to contain (the parts of) the features from the input and overlay layers that do not overlap features from the other layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Overlay fields prefix Pasirinktinis	`OVERLAY_FIELDS_`	[tekstas]	Add a prefix to identify fields of the overlay layer. Duplicate field names will be appended a count suffix to avoid collision.
Tinklelio dydis Pasirinktinis	`GRID_SIZE`	[skaičius: slankaus kablelio] Numatytas: jokio	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Symmetrical difference	`OUTPUT`	[toks pat, kaip įvestis]	Layer containing (parts of) features from each layer not overlapping the other layer, with both layers' attributes.

Pythono kodas

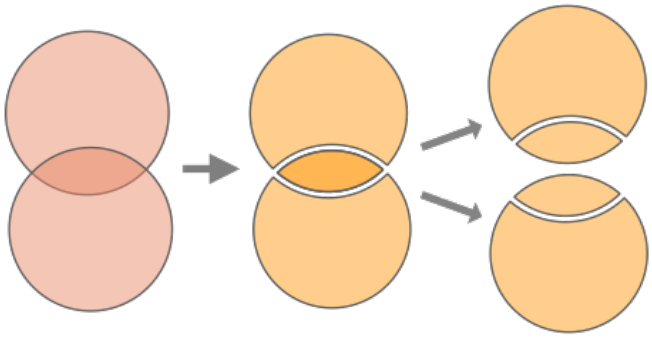
Algorithm ID: `native:symmetricaldifference`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

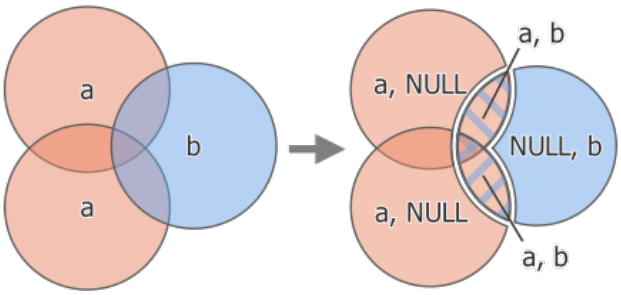
Union

Checks overlaps between features within the input layer and creates separate features for overlapping and non-overlapping parts. The area of overlap will create as many identical overlapping features as there are features that participate in that overlap.



24.159 Fig. : Union operation with a single input layer with two overlapping features (left) - resulting in four features (middle) - features moved for clarity (right)

An overlay layer can also be used, in which case features from each layer are split at their overlap with features from the other one, creating a layer containing all the portions from both input and overlay layers. Features on the same layer will not split each other. The attribute table of the union layer is filled with attribute values from the respective original layer for non-overlapping features, and attribute values from both layers for overlapping features.



24.160 Fig. : Union operation between a two-feature input layer ‚a' and a single feature overlay layer ‚b' (left) - resulting five-feature layer with attributes from both layers (right)

Pastaba

With an overlay layer, features on the same layer will not split each other. If you want to split overlaps on the same layer as well as other layers, first run the algorithm with multiple layers then run the algorithm again with only the previous output.

Numatytas meniu: *Vektorius ► Geometrijos įrankiai*

Įspėjimas

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Taip pat žiūrėkite

Union (multiple), *Apkirpti*, *Skirtumas*, *Intersection*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Input vector layer to split at any intersections.
Perdengimo sluoksnis Pasirinktinis	`OVERLAY`	[vektorius: geometrija]	Layer that will be combined to the first one. Ideally the geometry type should be the same as input layer.
Union	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the layer to contain the (split and duplicated) features from the input layer and the overlay layer. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Overlay fields prefix Pasirinktinis	`OVERLAY_FIELDS_`	[tekstas]	Add a prefix to identify fields of the overlay layer. Duplicate field names will be appended a count suffix to avoid collision.
Tinklelio dydis Pasirinktinis	`GRID_SIZE`	[skaičius: slankaus kablelio] Numatytas: jokio	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Union	`OUTPUT`	[toks pat, kaip įvestis]	Layer containing all the overlapping and non-overlapping parts from the processed layer(s).

Pythono kodas

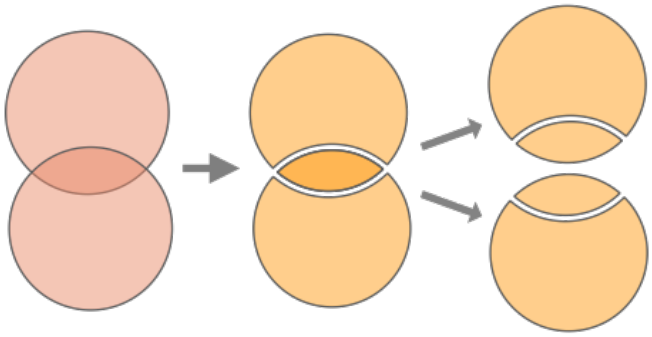
Algorithm ID: `native:union`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

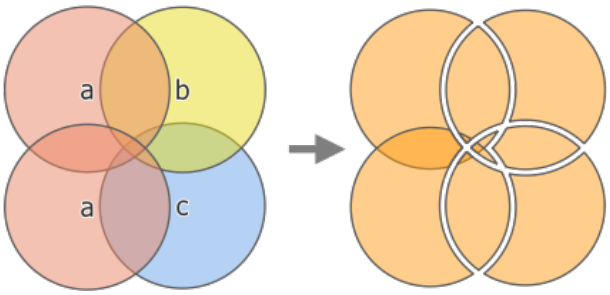
Union (multiple)

Checks overlaps between features within the input layer and creates separate features for overlapping and non-overlapping parts. The area of overlap will create as many identical overlapping features as there are features that participate in that overlap.



24.161 Fig. : Union operation with a single input layer with two overlapping features (left) - resulting in four features (middle) - features moved for clarity (right)

Multiple overlay layers can also be used, in which case features from each layer are split at their overlap with features from all other layers, creating a layer containing all the portions from both input and overlay layers. Features on the same layer will not split each other. The attribute table of the Union layer is filled with attribute values from the respective original layer for non-overlapping features, and attribute values from overlay layers for overlapping features.



24.162 Fig. : Union operation between a two-feature input layer ‚a‘ and single feature overlay layers ‚b‘ and ‚c‘ (left) - resulting eleven-feature layer with attributes from all layers (right)

Pastaba

With an overlay layer, features on the same layer will not split each other. If you want to split overlaps on the same layer as well as other layers, first run the algorithm with multiple layers then run the algorithm again with only the previous output.

Įspėjimas

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Taip pat žiūrėkite

Union, *Apkirpti*, *Skirtumas*, *Intersection*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Input vector layer to split at any intersections.
Perdengimo sluoksniai Pasirinktinis	`OVERLAYS`	[vektorius: geometrija] [sąrašas]	Layers that will be combined to the first one. Ideally the geometry type should be the same as input layer.
Union	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the layer to contain the (split and duplicated) features from the input layer and the overlay layers. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Overlay fields prefix Pasirinktinis	`OVERLAY_FIELDS_`	[tekstas]	Add a prefix to identify fields of the overlay layers. Duplicate field names will be appended a count suffix to avoid collision.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Union	`OUTPUT`	[toks pat, kaip įvestis]	Layer containing all the overlapping and non-overlapping parts from the processed layer(s), with all layers' attributes.

Pythono kodas

Algorithm ID: `native:multiunion`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.28 Vektorių parinkimas

Ištraukti pagal atributą

Iš įvesties sluoksnio sukuria du vektorinius sluoksnius: vieną tik su atitinkančiais geoobjektais ir antrą su neatitinkančiais geoobjektais.

Geoobjektų skirstymo į gaunamus sluoksnius kriterijus remiasi įvesties sluoksnio atributo reikšmėmis.

Taip pat žiūrėkite

Select by attribute

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Sluoksnis, iš kurio ištraukiami geoobjektai.
Parinkimo atributas	`FIELD`	[lentelės laukas: bet koks]	Sluoksnio filtravimo laukas
Veiksmas	`OPERATOR`	[sąrašas] Numatytas: 0	Yra daug skirtingų galimų veiksmų: • 0 — = • 1 — ≠ • 2 — >>br>• 3 — >= • 4 — < • 5 — <= • 6 — prasideda su • 7 — turi • 8 — yra null • 9 — nėra null • 10 — neturi
Reikšmė Pasirinktinis	`VALUE`	[tekstas]	Įvertinama reikšmė
Ištraukti (atributas)	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį atitinkantiems geoobjektams. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Ištraukti (neatitinkantys)	`FAIL_OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Praleisti išvestį]`	Nurodykite išvesties vektorinį sluoksnį neatitinkantiems geoobjektams. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Ištraukti (atributas)	`OUTPUT`	[toks pat, kaip įvestis]	Vektorinis sluoksnis su atitinkančiais geoobjektais iš įvesties sluoksnio
Ištraukti (neatitinkantys)	`FAIL_OUTPUT`	[toks pat, kaip įvestis]	Vektorinis sluoksnis su neatitinkančiais geoobjektais iš įvesties sluoksnio

Pythono kodas

Algoritmo ID: `native:extractbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Ištraukti pagal išraišką

Iš įvesties sluoksnio sukuria du vektorinius sluoksnius: vieną tik su atitinkančiais geoobjektais ir antrą su neatitinkančiais geoobjektais.

Geoobjektų pridėjimo į rezultato sluoksnį kriterijus remiasi QGIS išraiška. Daugiau informacijos apie išraiškas rasite skyriuje *Išraiškos*.

Taip pat žiūrėkite

Select by expression

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties vektorinis sluoksnis
Išraiška	`EXPRESSION`	[išraiška]	Vektorinio sluoksnio filtravimo išraiška
Atitinkantys geoobjektai	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį atitinkantiems geoobjektams. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Neatitinkantys	`FAIL_OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Praleisti išvestį]`	Nurodykite išvesties vektorinį sluoksnį neatitinkantiems geoobjektams. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Atitinkantys geoobjektai	`OUTPUT`	[toks pat, kaip įvestis]	Vektorinis sluoksnis su atitinkančiais geoobjektais iš įvesties sluoksnio
Neatitinkantys	`FAIL_OUTPUT`	[toks pat, kaip įvestis]	Vektorinis sluoksnis su neatitinkančiais geoobjektais iš įvesties sluoksnio

Pythono kodas

Algoritmo ID: `native:extractbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Ištraukti pagal vietą

Sukuria naują vektorinį sluoksnį, kuriame yra tik atitinkantys įvesties sluoksnio geoobjektai.

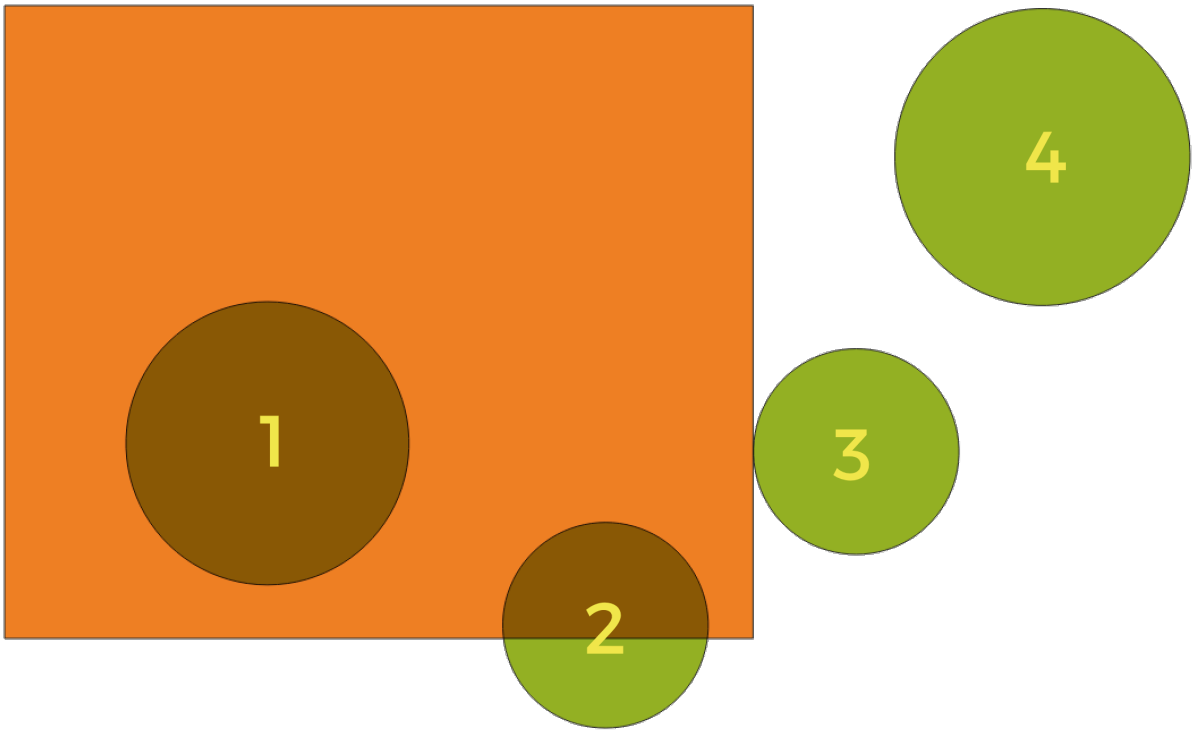
Geoobjektų pridėjimo į rezultatų sluoksnį kriterijus remiasi erdviniu ryšiu tarp kiekvieno geoobjekto ir geoobjektų papildomame sluoksnyje.

Taip pat žiūrėkite

Select by location, *Ištraukti pagal atstumą*

Erdvinių ryšių tyrinėjimas

Geometriniai predikatai yra funkcijos, kurios naudojamos nustatant, kokį erdvinį ryšį geoobjektas turi su kitu, lyginant ar ir kaip jų geometrijos dalinasi erdvės dalimi.



24.163 Fig. : Erdvinių ryšių tarp sluoksnių paieška

Naudojant aukščiau pateiktą iliustraciją, mes ieškome žalių apskritimų, erdviškai lygindami juos su oranžiniu stačiakampiu geoobjektus. Galimi geometriniai predikatai yra:

Susikerta
Tikrina, ar viena geometrija kerta kitą. Grąžina 1 (true), jei geometrijos erdviškai kertasi (dalinasi kažkokia erdvės dalimi - persidengia arba liečia) ir 0, jei to nėra. Iliustracijoje aukščiau True bus grąžinama apskritimams 1, 2 ir 3.

Įtraukia
Grąžina 1 (true) tada ir tik tada, jei nei vienas b taškas nėra už a ribų, ir bent vienas b vidaus taškas yra a viduje. Iliustracijoje nei vienas apskritimas nebūtų grąžintas, bet stačiakampis būtų, jei jūs ieškotumėte atvirkščiai, nes pilnai stačiakampyje yra 1 apskritimas. Tai yra priešingybė *yra viduje*.

Atskiras
Grąžina 1 (true), jei geometrijos nesidalina jokia erdvės dalimi (nėra persidengimo, nėra susilietimo). Grąžinamas tik 4 apskritimas.

Lygūs
Grąžina 1 (true), tada ir tik tada, jei geometrijos yra lygiai tokios pačios. Negrąžinami jokie apskritimai.

Liečia
Tikrina, ar geometrija liečia kitą geometriją. Grąžina 1 (true), jei geometrijos turi bent vieną bendrą tašką, bet jų vidus nepersidengia. Grąžinamas tik 3 apskritimas.

Persidengia
Tikrina, ar viena geometrija persidengia su kita. Grąžina 1 (true), jei geometrijos dalinasi erdve, turi tokius pačius matmenis, bet nei viena iš jų nėra pilnai kitoje. Grąžinamas tik 2 apskritimas.

Yra viduje
Tikrina, ar geometrija yra kitos viduje. Grąžina 1 (true), jei geometrija a yra pilnai geometrijos b viduje. Grąžinamas tik apskritimas 1.

Kerta
Grąžina 1 (true), jei pateiktos geometrijos turi kelis, bet ne visus, bendrus vidaus taškus ir jų susikirtimas turi mažiau matmenų nei daugiausiai matmenų turinti geometrija. Pavyzdžiui, linijos ir poligono susikirtimas bus

linija (true). Poligono ir poligono susikirtimas yra poligonas (false). Iliustracijoje nebūtų grąžinamas nei vienas apskritimas.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Ištraukti geoobjektus iš	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Kur geoobjektai (geometrinis predikatas)	`PREDICATE`	[enumeration] [list] Numatytas: [0]	Tipas erdvinio ryšio, kurį įvesties geoobjektas turi turėti su susikertančiu objektu, kad jis būtų parinktas. Vienas ar daugiau iš: • 0 — susikerta • 1 — įtraukia • 2 — atskiras • 3 — lygus • 4 — liečia • 5 — persidengia • 6 — yra viduje • 7 — kerta Jei parinkta daugiau nei viena sąlyga, bent viena iš jų (OR veiksmas) turi būti tenkinama, kad geoobjektas būtų ištrauktas.
Lyginant geoobjektus iš	`INTERSECT`	[vektorius: geometrija]	Susidengimo vektorinis sluoksnis
Ištrauktas (vieta)	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį geoobjektams, kurie turi pasirinktą erdvinį ryšį(ius) su vienu ar daugiau geobjektų iš palyginimo sluoksnio. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Ištrauktas (vieta)	`OUTPUT`	[toks pat, kaip įvestis]	Vektorinis sluoksnis su geoobjektais iš įvesties sluoksnio, kurie turi pasirinktą erdvinį ryšį(ius) su vienu ar daugiau geoobjektų iš palyginimo sluoksnio.

Pythono kodas

Algoritmo ID: `native:extractbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Ištraukti pagal atstumą

Sukuria naują vektorinį sluoksnį, kuriame yra tik atitinkantys įvesties sluoksnio geoobjektai. Geoobjektai nukopijuojami kai jie yra nurodytu atstumu nuo papildomo atskaitos sluoksnio geoobjektų.

Taip pat žiūrėkite

Select within distance, *Ištraukti pagal vietą*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Ištraukti geoobjektus iš	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis, iš kurio reikia kopijuoti geoobjektus
Lyginant geoobjektus iš	`REFERENCE`	[vektorius: geometrija]	Vektorinis sluoksnis, kurio geoobjektų artumas yra naudojamas
Kur geoobjektai yra arčiau nei	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 100.0	Maksimalus atstumas nuo atskaitos geoobjektų parenkant įvesties geoobjektus
Keisti dabartinį parinkimą	`METHOD`	[sąrašas] Numatytas: 0	Kaip valdyti algoritmo parinkimą. Vienas iš: • 0 — kurti naują parinkimą • 1 — pridėti į dabartinį parinkimą • 2 — parinkti dabartiniame parinkime • 3 — išimti iš dabartinio parinkimo
Ištrauktas (vieta)	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį geoobjektams, kurie yra arčiau nei nurodytas atstumas nuo atskaitos geoobjektų. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Ištrauktas (vieta)	`OUTPUT`	[toks pat, kaip įvestis]	Vektorinis sluoksnis su įvesties sluoksnio geoobjektais, kurie atitinka atstumo nuo atskaitos geoobjektų sąlygą

Pythono kodas

Algoritmo ID: `native:extractwithindistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Filter by geometry type

Filters features by their geometry type. Incoming features will be directed to different outputs based on whether they have a point, line or polygon geometry.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Layer to evaluate

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Point features Pasirinktinis	`POINTS`	[vector: point]	Layer with points
Line features Pasirinktinis	`LINES`	[vektorius: linija]	Layer with lines
Polygon features Pasirinktinis	`POLYGONS`	[vektorius: poligonas]	Layer with polygons
Features with no geometry Pasirinktinis	`NO_GEOMETRY`	[vektorius: lentelė]	Geometry-less vector layer

Pythono kodas

Algorithm ID: `native:filterbygeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Random extract

Takes a vector layer and generates a new one that contains only a subset of the features in the input layer.

The subset is defined randomly, based on feature IDs, using a percentage or count value to define the total number of features in the subset.

Įspėjimas

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Taip pat žiūrėkite

Random selection

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Source vector layer to select the features from
Metodas	`METHOD`	[sąrašas] Numatytas: 0	Random selection methods. One of: • 0 — Number of selected features • 1 — Percentage of selected features
Number/percentage of selected features	`NUMBER`	[skaičius: sveikas skaičius] Numatytas: 10	Number or percentage of features to select
Extracted (random)	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer for the randomly selected features. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Extracted (random)	`OUTPUT`	[toks pat, kaip įvestis]	Vector layer containing randomly selected features from the input layer

Pythono kodas

Algorithm ID: `native:randomextract`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Random extract within subsets

Takes a vector layer and generates a new one that contains only a subset of the features in the input layer.

The subset is defined randomly, based on feature IDs, using a percentage or count value to define the total number of features in the subset. The percentage/count value is not applied to the whole layer, but instead to each category. Categories are defined according to a given attribute.

Taip pat žiūrėkite

Random selection within subsets

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Vector layer to select the features from
ID field	`FIELD`	[lentelės laukas: bet koks]	Category of the source vector layer to select the features from
Metodas	`METHOD`	[sąrašas] Numatytas: 0	Random selection method. One of: • 0 — Number of selected features • 1 — Percentage of selected features
Number/percentage of selected features	`NUMBER`	[skaičius: sveikas skaičius] Numatytas: 10	Number or percentage of features to select

continues on next page

Table 24.801 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Extracted (random stratified)	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer for the randomly selected features. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Extracted (random stratified)	`OUTPUT`	[toks pat, kaip įvestis]	Vector layer containing randomly selected features from the input layer

Pythono kodas

Algorithm ID: `qgis:randomextractwithinsubsets`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Random selection

Takes a vector layer and selects a subset of its features. No new layer is generated by this algorithm.

The subset is defined randomly, based on feature IDs, using a percentage or count value to define the total number of features in the subset.

Numatytas meniu: *Vektorius* ► *Tyrimo įrankiai*

Taip pat žiūrėkite

Random extract

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Vector layer for the selection
Metodas	`METHOD`	[sąrašas] Numatytas: 0	Random selection method. One of: • 0 — Number of selected features • 1 — Percentage of selected features
Number/percentage of selected features	`NUMBER`	[skaičius: sveikas skaičius] Numatytas: 10	Number or percentage of features to select

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[toks pat, kaip įvestis]	The input layer with features selected

Pythono kodas

Algorithm ID: `qgis:randomselection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Random selection within subsets

Takes a vector layer and selects a subset of its features. No new layer is generated by this algorithm.

The subset is defined randomly, based on feature IDs, using a percentage or count value to define the total number of features in the subset.

The percentage/count value is not applied to the whole layer, but instead to each category.

Categories are defined according to a given attribute, which is also specified as an input parameter for the algorithm.

No new outputs are created.

Numatytas meniu: *Vektorius* ► *Tyrimo įrankiai*

Taip pat žiūrėkite

Random extract within subsets

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Vector layer to select features in
ID field	`FIELD`	[lentelės laukas: bet koks]	Category of the input layer to select the features from
Metodas	`METHOD`	[sąrašas] Numatytas: 0	Random selection method. One of: • 0 — Number of selected features • 1 — Percentage of selected features
Number/percentage of selected features	`NUMBER`	[skaičius: sveikas skaičius] Numatytas: 10	Number or percentage of features to select

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[toks pat, kaip įvestis]	The input layer with features selected

Pythono kodas

Algorithm ID: `qgis:randomselectionwithinsubsets`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Select by attribute

Creates a selection in a vector layer.

The criteria for selecting features is based on the values of an attribute from the input layer.

Taip pat žiūrėkite

Ištraukti pagal atributą

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Vector layer to select features in
Parinkimo atributas	`FIELD`	[lentelės laukas: bet koks]	Sluoksnio filtravimo laukas
Veiksmas	`OPERATOR`	[sąrašas] Numatytas: 0	Yra daug skirtingų galimų veiksmų: • 0 — = • 1 — ≠ • 2 — >> • 3 — >= • 4 — < • 5 — <= • 6 — prasideda su • 7 — turi • 8 — yra null • 9 — nėra null • 10 — neturi
Reikšmė Pasirinktinis	`VALUE`	[tekstas]	Įvertinama reikšmė
Keisti dabartinį parinkimą	`METHOD`	[sąrašas] Numatytas: 0	Kaip valdyti algoritmo parinkimą. Vienas iš: • 0 — kurti naują parinkimą • 1 — pridėti į dabartinį parinkimą • 2 — removing from current selection • 3 — selecting within current selection

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[toks pat, kaip įvestis]	The input layer with features selected

Pythono kodas

Algorithm ID: `qgis:selectbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Select by expression

Creates a selection in a vector layer.

The criteria for selecting features is based on a QGIS expression. For more information about expressions see the *Išraiškos*.

Taip pat žiūrėkite

Ištraukti pagal išraišką

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties vektorinis sluoksnis
Išraiška	`EXPRESSION`	[išraiška]	Expression to filter the input layer
Keisti dabartinį parinkimą	`METHOD`	[sąrašas] Numatytas: 0	Kaip valdyti algoritmo parinkimą. Vienas iš: • 0 — kurti naują parinkimą • 1 — pridėti į dabartinį parinkimą • 2 — removing from current selection • 3 — selecting within current selection

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[toks pat, kaip įvestis]	The input layer with features selected

Pythono kodas

Algorithm ID: `qgis:selectbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Select by location

Creates a selection in a vector layer.

The criteria for selecting features is based on the spatial relationship between each feature and the features in an additional layer.

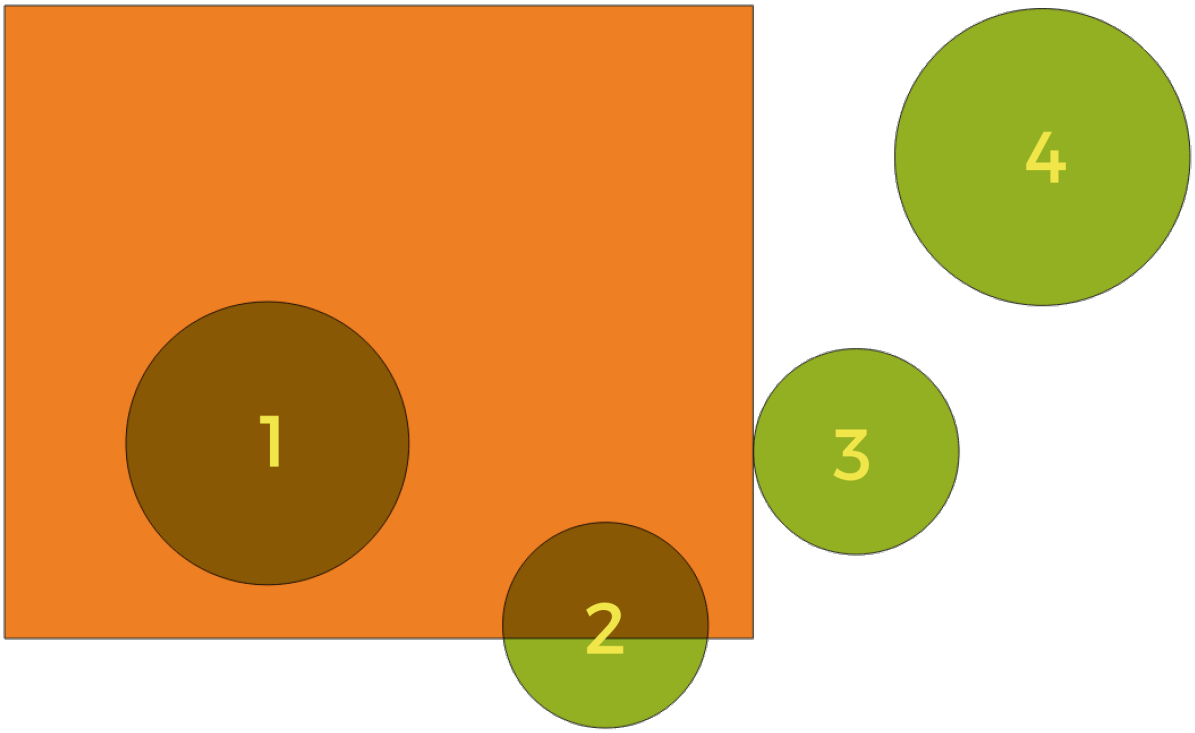
Numatytas meniu: *Vektorius* ► *Tyrimo įrankiai*

Taip pat žiūrėkite

Ištraukti pagal vietą, *Select within distance*

Erdvinių ryšių tyrinėjimas

Geometriniai predikatai yra funkcijos, kurios naudojamos nustatant, kokį erdvinį ryšį geoobjektas turi su kitu, lyginant ar ir kaip jų geometrijos dalinasi erdvės dalimi.



24.164 Fig. : Erdvinių ryšių tarp sluoksnių paieška

Naudojant aukščiau pateiktą iliustraciją, mes ieškome žalių apskritimų, erdviškai lygindami juos su oranžiniu stačiakampiu geoobjektus. Galimi geometriniai predikatai yra:

Susikerta
: Tikrina, ar viena geometrija kerta kitą. Grąžina 1 (true), jei geometrijos erdviškai kertasi (dalinasi kažkokia erdvės dalimi - persidengia arba liečia) ir 0, jei to nėra. Iliustracijoje aukščiau True bus grąžinama apskritimams 1, 2 ir 3.

Įtraukia
: Grąžina 1 (true) tada ir tik tada, jei nei vienas b taškas nėra už a ribų, ir bent vienas b vidaus taškas yra a viduje. Iliustracijoje nei vienas apskritimas nebūtų grąžintas, bet stačiakampis būtų, jei jūs ieškotumėte atvirkščiai, nes pilnai stačiakampyje yra 1 apskritimas. Tai yra priešingybė *yra viduje*.

Atskiras
Grąžina 1 (true), jei geometrijos nesidalina jokia erdvės dalimi (nėra persidengimo, nėra susilietimo). Grąžinamas tik 4 apskritimas.

Lygūs
Grąžina 1 (true), tada ir tik tada, jei geometrijos yra lygiai tokios pačios. Negrąžinami jokie apskritimai.

Liečia
Tikrina, ar geometrija liečia kitą geometriją. Grąžina 1 (true), jei geometrijos turi bent vieną bendrą tašką, bet jų vidus nepersidengia. Grąžinamas tik 3 apskritimas.

Persidengia
Tikrina, ar viena geometrija persidengia su kita. Grąžina 1 (true), jei geometrijos dalinasi erdve, turi tokius pačius matmenis, bet nei viena iš jų nėra pilnai kitoje. Grąžinamas tik 2 apskritimas.

Yra viduje
Tikrina, ar geometrija yra kitos viduje. Grąžina 1 (true), jei geometrija a yra pilnai geometrijos b viduje. Grąžinamas tik apskritimas 1.

Kerta
Grąžina 1 (true), jei pateiktos geometrijos turi kelis, bet ne visus, bendrus vidaus taškus ir jų susikirtimas turi mažiau matmenų nei daugiausiai matmenų turinti geometrija. Pavyzdžiui, linijos ir poligono susikirtimas bus linija (true). Poligono ir poligono susikirtimas yra poligonas (false). Iliustracijoje nebūtų grąžinamas nei vienas apskritimas.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Select features from	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Kur geoobjektai (geometrinis predikatas)	`PREDICATE`	[enumeration] [list] Numatytas: [0]	Tipas erdvinio ryšio, kurį įvesties geoobjektas turi turėti su susikertančiu objektu, kad jis būtų parinktas. Vienas ar daugiau iš: • 0 — susikerta • 1 — įtraukia • 2 — atskiras • 3 — lygus • 4 — liečia • 5 — persidengia • 6 — yra viduje • 7 — kerta Jei parinkta daugiau nei viena sąlyga, bent viena iš jų (OR veiksmas) turi būti tenkinama, kad geoobjektas būtų ištrauktas.
Lyginant geoobjektus iš	`INTERSECT`	[vektorius: geometrija]	Susidengimo vektorinis sluoksnis
Keisti dabartinį parinkimą	`METHOD`	[sąrašas] Numatytas: 0	Kaip valdyti algoritmo parinkimą. Vienas iš: • 0 — kurti naują parinkimą • 1 — pridėti į dabartinį parinkimą • 2 — parinkti dabartiniame parinkime • 3 — išimti iš dabartinio parinkimo

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[toks pat, kaip įvestis]	The input layer with features selected

Pythono kodas

Algorithm ID: `native:selectbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Select within distance

creates a selection in a vector layer. Features are selected wherever they are within the specified maximum distance from the features in an additional reference layer.

Taip pat žiūrėkite

Ištraukti pagal atstumą, *Select by location*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Select features from	`INPUT`	[vektorius: geometrija]	Input vector layer to select features from
Lyginant geoobjektus iš	`REFERENCE`	[vektorius: geometrija]	Vektorinis sluoksnis, kurio geoobjektų artumas yra naudojamas
Kur geoobjektai yra arčiau nei	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 100.0	The maximum distance around reference features to select input features
Keisti dabartinį parinkimą	`METHOD`	[sąrašas] Numatytas: 0	Kaip valdyti algoritmo parinkimą. Vienas iš: • 0 — kurti naują parinkimą • 1 — pridėti į dabartinį parinkimą • 2 — parinkti dabartiniame parinkime • 3 — išimti iš dabartinio parinkimo

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[toks pat, kaip įvestis]	The input layer with features selected

Pythono kodas

Algorithm ID: `native:selectwithindistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.29 Vektorinė lentelė

Pridėti automatiškai didinamą lauką

Prideda į vektorinį sluoksnį naują sveiko skaičiaus lauką, kuriame kiekvienam geoobjektui reikšmė imama iš sekos.

Šį lauką galima naudoti kaip sluoksnio geoobjektų unikalų ID. Naujas atributas nepridedamas į įvesties sluoksnį, o vietoje to kuriamas naujas sluoksnis.

Galima nurodyti pradinę reikšmę, nuo kurios prasideda seka. Pasirinktinai, didėjanti seka gali remtis laukų grupavimu, taipogi galima nurodyti geoobjektų rikiavimo tvarką.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties vektorinis sluoksnis.
Lauko pavadinimas	`FIELD_NAME`	[tekstas] Numatytas: „AUTO“	Lauko su automatiškai didinama reikšme pavadinimas
Reikšmes pradėti nuo Pasirinktinis	`START`	[skaičius: sveikas skaičius] Numatytas: 0	Parinkite pradinį didėjančios sekos skaičių
Modulio reikšmė Pasirinktinis	`MODULUS`	[skaičius: sveikas skaičius] Numatytas: 0	Nurodžius neprivalomą modulį, reikšmės bus pradedamos iš naujo nuo START reikšmės, kai lauko reikšmė pasieks modulio reikšmę. `0` reiškia nepradėti iš naujo.
Grupuoti reikšmes pagal Pasirinktinis	`GROUP_FIELDS`	[lentelės laukas: bet koks] [sąrašas]	Parinkite grupuojamą lauką(us): vietoje vieno skaičiavimo visam sluoksniui, atskiras skaitliukas apdorojamas kiekvienai reikšmei, kurią grąžina šių laukų kombinacija.

continues on next page

Table 24.815 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Rikiavimo išraiška Pasirinktinis	SORT_EXPRESSIO	[išraiška]	Naudoti išraišką geoobjektų rikiavimui sluoksnyje globaliai arba, jei nurodyta, pagal grupavimo laukus.
Rikiuoti didėjančiai	SORT_ASCENDING	[boolean] Numatytas: Taip	Nurodžius rikiuoti didėjančiai, naudokite šią parinktį, kad valdytumėte rikiuotę, kuria geoobjektams priskiriamos reikšmės.
Null reikšmės rikiuotės pradžioje	SORT_NULLS_FIR	[boolean] Numatytas: ne	Nurodžius rikiuotės išraišką, naudokite šią parinktį, kad *Null* reikšmės būtų pirmos ar paskutinės.
Padidintos	OUTPUT	[toks pat, kaip įvestis] Numatytas: [Kurti laikiną sluoksnį]	Nurodykite išvesties vektorinį sluoksnį su automatiškai didinamos reikšmės lauku. *Vienas iš*: • Kurti laikiną sluoksnį (TEMPORARY_OUTPUT) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Padidintos	OUTPUT	[toks pat, kaip įvestis]	Vektorinis sluoksnis su automatiškai didėjančiu lauku

Pythono kodas

Algoritmo ID: native:addautoincrementalfield

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Pridėti lauką į atributų lentelę

Prideda į vektorinį sluoksnį naują lauką.

Atributo pavadinimas ir charakteristikos nurodomos kaip parametrai.

Naujas atributas nėra pridedamas į įvesties sluoksnį, vietoje to kuriamas naujas sluoksnis.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties sluoksnis
Lauko pavadinimas	`FIELD_NAME`	[tekstas]	Naujo lauko pavadinimas
Lauko tipas	`FIELD_TYPE`	[sąrašas] Numatytas: 0	Naujo lauko tipas. Galite pasirinkti vieną iš: • 0 — Sveikas skaičius (32 bit) • 1 — Dešimtainis (double) • 2 — Tekstas (string) • 3 — Loginis • 4 — Data • 5 — Laikas • 6 — Data ir laikas • 7 — Dvejetainis objektas (BLOB) • 8 — Eilučių sąrašas • 9 — Sveikų skaičių sąrašas • 10 — Dešimtainių (double) sąrašas
Lauko ilgis	`FIELD_LENGTH`	[skaičius: sveikas skaičius] Numatytas: 10	Lauko ilgis
Lauko tikslumas	`FIELD_PRECISIO`	[skaičius: sveikas skaičius] Numatytas: 0	Lauko tikslumas. Naudojama slankaus kablelio lauko tipui.
Lauko pseudonimas Pasirinktinis	`FIELD_ALIAS`	[tekstas]	Nurodykite pavadinimą, kuris turėtų būti naudojamas kaip lauko pseudonimas. Palaikoma ne visiems formatams.
Lauko komentaras Pasirinktinis	`FIELD_COMMENT`	[tekstas]	Įrašyti šį lauką aprašantį komentarą. Palaikoma ne visiems formatams.
Papildytas	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildytas	`OUTPUT`	[toks pat, kaip įvestis]	Vektorinis sluoksnis su pridėti nauju lauku

Pythono kodas

Algoritmo ID: `native:addfieldtoattributestable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Pridėti unikalios reikšmės indekso lauką

Nurodytam vektoriniam sluoksniui ir atributui prideda naują skaičiaus lauką.

Šio lauko reikšmės atitinka nurodyto atributo reikšmes, taigi geoobjektai su tokia pačia atributo reikšme turės tą pačią reikšmę naujame skaičiaus lauke.

Tai sukuria skaitmeninį nurodyto atributo ekvivalentą, kuris apibrėžia tas pačias klases.

Naujas atributas nėra pridedamas į įvesties sluoksnį, vietoje to kuriamas naujas sluoksnis.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties sluoksnis.
Klasės laukas	`FIELD`	[tablefield: any]	Geoobjektasi, kurie turi tą pačią šio lauko reikšmę turės tą patį indeksą.
Išvesties lauko pavadinimas	`FIELD_NAME`	[tekstas] Numatytas: ‚NUM_FIELD‘	Naujo lauko su indeksu pavadinimas.
Sluoksnis su indekso lauku	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Vektorinis sluoksnis su skaitmeniniu lauku turinčiu indeksus. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.
Klasės suvestinė	`SUMMARY_OUTPUT`	[vektorius: lentelė] Numatytas: `[Praleisti išvestį]`	Nurodykite lentelę, kuri turės klasės lauko suvestinę susietą su atitinkama unikalia reikšme. *Vienas iš*: • Praleisti išvestį • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sluoksnis su indekso lauku	`OUTPUT`	[toks pat, kaip įvestis]	Vektorinis sluoksnis su skaitiniu lauku turinčiu indeksus.
Klasės suvestinė	`SUMMARY_OUTPUT`	[vektorius: lentelė]	Lentelė su klasės lauko suvestine susieta su atitinkama unikalia reikšme.

Pythono kodas

Algoritmo ID: `native:adduniquevalueindexfield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Add X/Y fields to layer

Adds X and Y (or latitude/longitude) fields to a point layer. The X/Y fields can be calculated in a different CRS to the layer (e.g. creating latitude/longitude fields for a layer in a projected CRS).

Leidžia taškų *geoobjektus keisti vietoje*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vector: point]	Įvesties sluoksnis.
Coordinate system	`CRS`	[crs] Default: „EPSG:4326“	Coordinate reference system to use for the generated x and y fields.
Field prefix Pasirinktinis	`PREFIX`	[tekstas]	Prefix to add to the new field names to avoid name collisions with fields in the input layer.
Added fields	`OUTPUT`	[vector: point] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Added fields	`OUTPUT`	[vector: point]	The output layer - identical to the input layer but with two new double fields, `x` and `y`.

Pythono kodas

Algorithm ID: `native:addxyfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Advanced Python field calculator

Adds a new attribute to a vector layer, with values resulting from applying an expression to each feature.

The expression is defined as a Python function.

> **Įspėjimas**
>
> This algorithm is a potential security risk if executed with unchecked inputs, and may result in system damage or data leaks.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties vektorinis sluoksnis
Result field name	`FIELD_NAME`	[tekstas] Default: ‚NewField‘	Naujo lauko pavadinimas
Lauko tipas	`FIELD_TYPE`	[sąrašas] Numatytas: 0	Type of the new field. One of: • 0 — Sveikas skaičius (32 bit) • 1 — Dešimtainis (double) • 2 — Tekstas (string) • 3 — Loginis • 4 — Data • 5 — Laikas • 6 — Data ir laikas • 7 — Dvejetainis objektas (BLOB) • 8 — Eilučių sąrašas • 9 — Sveikų skaičių sąrašas • 10 — Dešimtainių (double) sąrašas

continues on next page

Table 24.823 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Lauko ilgis	`FIELD_LENGTH`	[skaičius: sveikas skaičius] Numatytas: 10	Lauko ilgis
Lauko tikslumas	`FIELD_PRECISIO`	[skaičius: sveikas skaičius] Numatytas: 3	Lauko tikslumas. Naudojama slankaus kablelio lauko tipui.
Global expression Pasirinktinis	`GLOBAL`	[tekstas]	The code in the global expression section will be executed only once before the calculator starts iterating through all the features of the input layer. Therefore, this is the correct place to import necessary modules or to calculate variables that will be used in subsequent calculations.
Formula	`FORMULA`	[tekstas]	The Python formula to evaluate. Example: To calculate the area of an input polygon layer you can add: `value = $geom.area()` To access an existing field use its name enclosed in the angle brackets. For example, to uppercase values of the „address“ field: `value = <address>.upper()`
Calculated	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the vector layer with the new calculated field. One of: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Calculated	`OUTPUT`	[toks pat, kaip įvestis]	Vector layer with the new calculated field

Pythono kodas

Algorithm ID: `qgis:advancedpythonfieldcalculator`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Drop field(s)

Takes a vector layer and generates a new one that has the same features but without the selected columns.

Taip pat žiūrėkite

Retain fields

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Input vector layer to drop field(s) from
Fields to drop	`COLUMN`	[lentelės laukas: bet koks] [sąrašas]	The field(s) to drop
Remaining fields	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specify the output vector layer with the remaining fields. *One of*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Remaining fields	`OUTPUT`	[toks pat, kaip įvestis]	Vector layer with the remaining fields

Pythono kodas

Algorithm ID: `native:deletecolumn`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Explode HStore Field

Creates a copy of the input layer and adds a new field for every unique key in the HStore field.

The expected field list is an optional comma separated list. If this list is specified, only these fields are added and the HStore field is updated. By default, all unique keys are added.

The PostgreSQL HStore is a simple key-value store used in PostgreSQL and GDAL (when reading an OSM file with the `other_tags` field.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
HStore field	`FIELD`	[tablefield: any]	The field(s) to drop
Expected list of fields separated by a comma Pasirinktinis	`EXPECTED_FIELD`	[tekstas] Numatytas: ‚'	Comma-separated list of fields to extract. The HStore field will be updated by removing these keys.
Exploded	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties vektorinį sluoksnį. Vienas iš: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Exploded	`OUTPUT`	[toks pat, kaip įvestis]	Output vector layer

Pythono kodas

Algorithm ID: `native:explodehstorefield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Extract binary field

Extracts contents from a binary field, saving them to individual files. Filenames can be generated using values taken from an attribute in the source table or based on a more complex expression.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Input vector layer containing the binary data
Binary field	`FIELD`	[tablefield: any]	Field containing the binary data
File name	`FILENAME`	[išraiška]	Field or expression-based text to name each output file
Destination folder	`FOLDER`	[aplankas] Numatytas: `[Įrašyti į laikiną aplanką]`	Folder in which to store the output files. *One of*: • Įrašyti į laikiną aplanką • Įrašyti į aplanką

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Folder	`FOLDER`	[aplankas]	The folder that contains the output files.

Pythono kodas

Algorithm ID: `native:extractbinary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Field calculator

Opens the field calculator (see *Išraiškos*). You can use all the supported expressions and functions.

A new layer is created with the result of the expression.

The field calculator is very useful when used in *Modelių dizaineris*.

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	The layer to calculate on
Išvesties lauko pavadinimas	`FIELD_NAME`	[tekstas]	The name of the field for the results
Output field type	`FIELD_TYPE`	[sąrašas] Numatytas: 0	The type of the field. One of: • 0 — Decimal (double) • 1 — Integer (32 bit) • 2 — Tekstas (string) • 3 — Date • 4 — Time • 5 — Date & Time • 6 — Boolean • 7 — Dvejetainis objektas (BLOB) • 8 — Eilučių sąrašas • 9 — Sveikų skaičių sąrašas • 10 — Dešimtainių (double) sąrašas
Output field width	`FIELD_LENGTH`	[skaičius: sveikas skaičius] Numatytas: 0	The length of the result field (minimum 0)
Lauko tikslumas	`FIELD_PRECISIO`	[skaičius: sveikas skaičius] Numatytas: 0	The precision of the result field (minimum 0, maximum 15)
Create new field	`NEW_FIELD`	[boolean] Numatytas: Taip	Should the result field be a new field
Formula	`FORMULA`	[išraiška]	The formula to use to calculate the result
Calculated	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Specification of the output layer. • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Calculated	`OUTPUT`	[toks pat, kaip įvestis]	Output layer with the calculated field values

Pythono kodas

Algorithm ID: `native:fieldcalculator`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Refactor fields

Allows editing the structure of the attribute table of a vector layer.

Fields can be modified in their type and name, using a fields mapping.

The original layer is not modified. A new layer is generated, which contains a modified attribute table, according to the provided fields mapping.

Pastaba

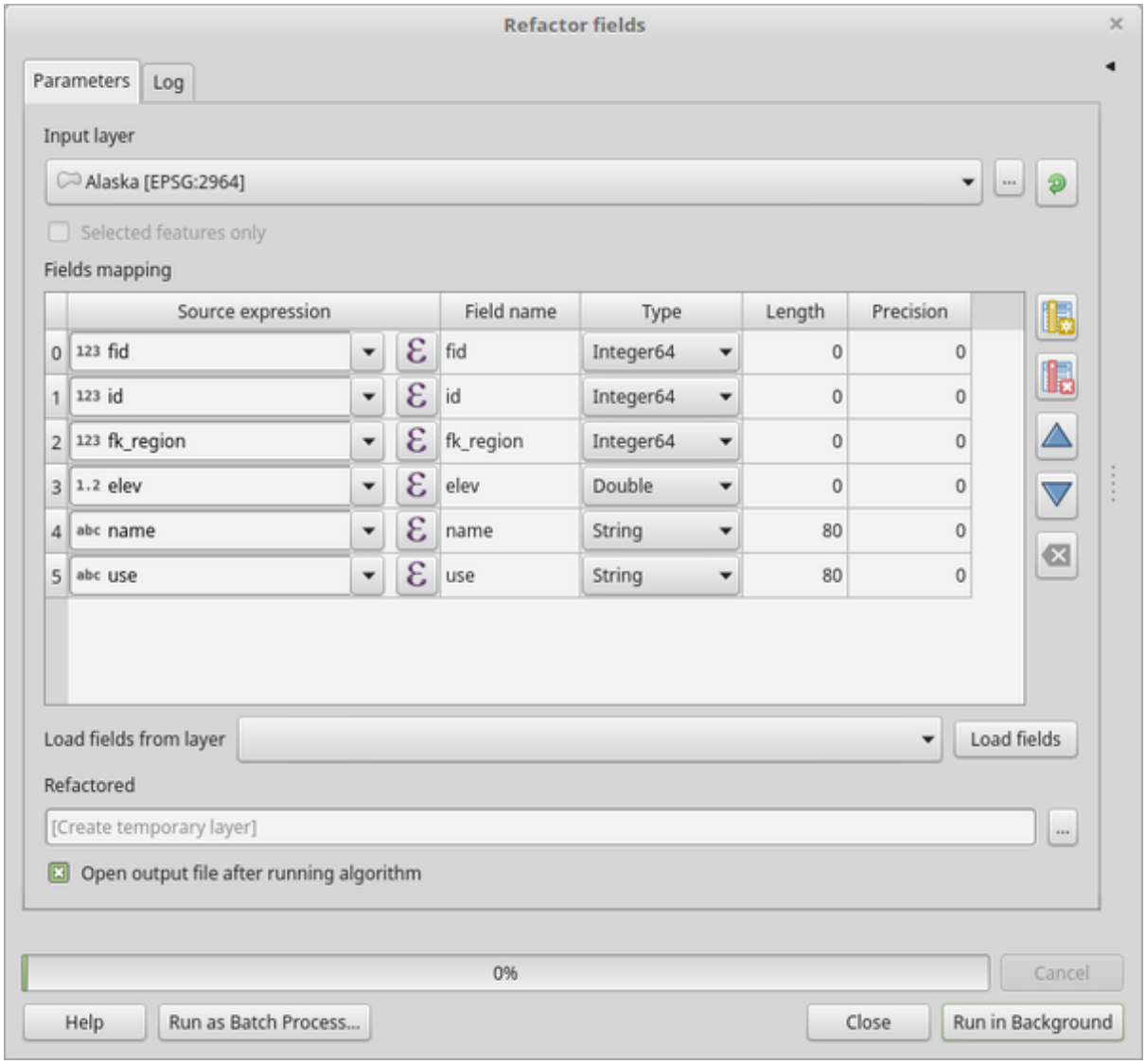
When using a template layer with *constraints* on fields, the information is displayed in the widget with a coloured background and tooltip. Treat this information as a hint during configuration. No constraints will be added on an output layer nor will they be checked or enforced by the algorithm.

The Refactor fields algorithm allows to:

- Change field names and types
- Add and remove fields
- Reorder fields
- Calculate new fields based on expressions
- Load field list from another layer

Pastaba

Šis algoritmas naudoja elipse paremtus skaičiavimus ir atsižvelgia į dabartinius *elipsoido nustatymus*.



24.165 Fig. : Refactor fields dialog

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	The layer to modify

continues on next page

Table 24.833 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Fields mapping	`FIELDS_MAPPING`	[sąrašas]	List of output fields with their definitions. The embedded table lists all the fields of the source layer and allows you to edit them: • Click to create a new field. • Click to remove a field. • Use and to change the selected field order. • Click to reset to the default view. • Click to invert the selection in the fields list. For each of the fields you'd like to reuse, you need to fill the following options: ***Source expression*** **(`expression`) [expression]** Įvesties sluoksnio lauką ar išraišką. ***Name*** **(`name`) [string]** Name of the field in the output layer. By default input field name is kept. ***Tipas*** **(`type`) [iš sąrašo]** Išvesties lauko duomenų tipas. Galimi tipai gali būti nesuderinami su išvesties sluoksnio tiekėju. Vienas iš: **Dėmesio** Kai kuriems tipams, pvz. sąrašams, papildomas parametras `sub_type` leidžia tiksliau apibūdinti konkretų duomenų tipą. Jis automatiškai nustatomas GUI, bet gali būti naudingas, jei jūs paleidžiate algoritmą Python'e arba komandinėje eilutėje. • 1 — Boolean • 2 — Integer (32bit) • 4 — Integer (64bit) • 6 — Decimal (double) • 9 — Integer list • 9 — Integer (64bit) list • 9 — Decimal (double) list • 10 — Text (string) • 11 — Eilučių sąrašas • 12 — Dvejetainis objektas (BLOB) • 14 — Data • 15 — Laikas • 16 — Data ir laikas ***Potipis*** **(`sub_type`) [iš sąrašo]** Kai kuriems laukų tipams, pvz. sąrašams, šis parametras padeda geriau apibrėžti kai kurių duomenų `type`. Jis automatiškai nustatomas GUI, bet gali būti naudingas, jei jūs paleidžiate algoritmą Python'e arba komandinėje eilutėje. *Vienas iš*: • 2 — Sveikų skaičių sąrašas • 4 — Sveikų skaičių (64bit) sąrašas

Table 24.833 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Refactored	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Išvesties sluoksnio specifikacija. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Refactored	`OUTPUT`	[toks pat, kaip įvestis]	Output layer with refactored fields

Pythono kodas

Algorithm ID: `native:refactorfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Rename field

Renames an existing field from a vector layer.

The original layer is not modified. A new layer is generated where the attribute table contains the renamed field.

Taip pat žiūrėkite

Refactor fields

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties vektorinis sluoksnis
Field to rename	`FIELD`	[tablefield: any]	The field to be altered
New field name	`NEW_NAME`	[tekstas]	The new field name

continues on next page

Table 24.835 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Renamed	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Išvesties sluoksnio specifikacija. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Renamed	`OUTPUT`	[toks pat, kaip įvestis]	Output layer with the renamed field

Pythono kodas

Algorithm ID: `native:renametablefield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Retain fields

Takes a vector layer and generates a new one that retains only the selected fields. All other fields will be dropped.

Taip pat žiūrėkite

Drop field(s)

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties vektorinis sluoksnis
Fields to retain	`FIELDS`	[lentelės laukas: bet koks] [sąrašas]	List of fields to keep in the layer

continues on next page

Table 24.837 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Retained fields	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Išvesties sluoksnio specifikacija. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Retained fields	`OUTPUT`	[toks pat, kaip įvestis]	Output layer with the retained fields

Pythono kodas

Algorithm ID: `native:retainfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Text to float

Modifies the type of a given attribute in a vector layer, converting a text attribute containing numeric strings into a numeric attribute (e.g. ‚1‘ to `1.0`).

The algorithm creates a new vector layer so the source one is not modified.

If the conversion is not possible the selected column will have `NULL` values.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties vektorinis sluoksnis.
Text attribute to convert to float	`FIELD`	[lentelės laukas: tekstas]	The string field for the input layer that is to be converted to a float field.

continues on next page

Table 24.839 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Float from text	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Kurti laikiną sluoksnį]`	Nurodykite išvesties sluoksnį. *Vienas iš*: • Kurti laikiną sluoksnį (`TEMPORARY_OUTPUT`) • Įrašyti į failą… • Įrašyti į geopackage… • Įrašyti į duombazės lentelę… • Pridėti į sluoksnį… Čia taipogi galima pakeisti failo koduotę.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Float from text	`OUTPUT`	[toks pat, kaip įvestis]	Output vector layer with the string field converted into a float field

Pythono kodas

Algorithm ID: `qgis:texttofloat`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.1.30 Vektorinės kaladėlės

Atsiųsti vektorines kaladėles

Atsiunčia įvesties vektorinių kaladėlių sluoksnio vektorines kaladėles ir įrašo jas į vietinį vektorinių kaladėlių failą.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorinės kaladėlės]	Vektorinių kaladėlių sluoksnis, kurio kaladėles reikia ištraukti

continues on next page

Table 24.841 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Apimtis	`EXTENT`	[apimtis]	Nurodykite atsiunčiamo ploto erdvinę apimtį. Ji viduje bus praplėsta iki kaladėlės dydžio kartotinio. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Maksimalus atsiunčiamas mastelio lygis	`MAX_ZOOM`	[skaičius: sveikas skaičius] Numatytas: 10	Nurodo kiek giliai reikia siųsti kaladėlių duomenis
Kaladėlių riba	`TILE_LIMIT`	[skaičius: sveikas skaičius] Numatytas: 100	Maksimalus atsiunčiamų kaladėlių skaičius, atsižvelgiant į mastelius ir apimtį.
Išvestis	`OUTPUT`	[vektorinės kaladėlės] Numatytas: [įrašyti į laikiną failą]	Išvesties vektorinių kaladėlių failo specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvestis	`OUTPUT`	[vektorinės kaladėlės]	Vietinis vektorinių kaladėlių failas, į kurį įrašytos atsiųstos kaladėlės.

Pythono kodas

Algoritmo ID: `native:downloadvectortiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Rašyti vektorines kaladėles (MBTiles)

Eksportuoja vieną ar daugiau vektorinių sluoksnį į vektorines kaladėles, duomenų formatą, optimizuotą greitam žemėlapio braižymui ir mažam duomenų kiekiui.

MBTiles yra specifikacija, apibrėžianti kaladėlėmis suskirstytų žemėlapio duomenų saugojimą SQLite duomenų bazėse momentiniam naudojimui ir perdavimui. MBTiles failai žinomi kaip kaladėlių aibės - „tilesets“.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`INPUT`	[vektorius: bet koks] [sąrašas]	Sąrašas sluoksnių, kuriuos reikia sudėti į kuriamas vektorines kaladėles
Minimalus mastelis	`MIN_ZOOM`	[skaičius: sveikas skaičius] Numatytas: 0	Mažiausias mastelis, kuriam teikiami kaladėlių duomenys. Nustatykite tarp 0 ir 24.
Maksimalus mastelis	`MAX_ZOOM`	[skaičius: sveikas skaičius] Numatytas: 3	Didžiausias mastelis, kuriam teikiami kaladėlių duomenys. Nustatykite tarp 0 ir 24.
Apimtis Pasirinktinis	`EXTENT`	[apimtis] Numatytas: jokio	Maksimali braižomo žemėlapio ploto apimtis. Ribos turi apibrėžti visų mastelių dengiamą plotą.
Metaduomenys: pavadinimas Pasirinktinis	`META_NAME`	[tekstas]	Kaladėlių rinkinio pavadinimas
Metaduomenys: aprašymas Pasirinktinis	`META_DESCRIPTI`	[tekstas]	Kaladėlių rinkinio turinio aprašymas
Metaduomenys: autoriai Pasirinktinis	`META_ATTRIBUTI`	[tekstas]	Autorių tekstas, kuris aprašo duomenų ir/arba žemėlapio stiliaus šaltinį.
Metaduomenys: versija Pasirinktinis	`META_VERSION`	[tekstas]	Kaladėlių rinkinio versija. Tai nusako paties kaladėlių rinkinio versiją, o ne MBTiles specifikaciją.
Metaduomenys: tipas Pasirinktinis	`META_TYPE`	[tekstas]	Kaladėlių rinkinio tipas. Galimos reikšmės yra `overlay` ar `baselayer`.
Metaduomenys: centras Pasirinktinis	`META_CENTER`	[tekstas]	Numatytojo žemėlapio vaizdo centras (kableliais atskirta skaičių eilutė: ilguma, platuma ir mastelis) . Pavyzdžiui: `-122.1906,37.7599,11`
Paskirties MBTiles	`OUTPUT`	[vektorinės kaladėlės] Numatytas: [įrašyti į laikiną failą]	Išvesties MBTiles failo specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Paskirties MBTiles	`OUTPUT`	[file]	Išvesties vektorinių kaladėlių `.mbtiles` failas.

Pythono kodas

Algoritmo ID: `native:writevectortiles_mbtiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Write vector tiles (XYZ)

Eksportuoja vieną ar daugiau vektorinių sluoksnį į vektorines kaladėles, duomenų formatą, optimizuotą greitam žemėlapio braižymui ir mažam duomenų kiekiui.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
File template	`XYZ_TEMPLATE`	[tekstas] Default: ‚{z}/{x}/{y}.pbf‘	Template to generate the vector tiles url
Įvesties sluoksniai	`INPUT`	[vektorius: bet koks] [sąrašas]	Sąrašas sluoksnių, kuriuos reikia sudėti į kuriamas vektorines kaladėles
Minimalus mastelis	`MIN_ZOOM`	[skaičius: sveikas skaičius] Numatytas: 0	Mažiausias mastelis, kuriam teikiami kaladėlių duomenys. Nustatykite tarp 0 ir 24.
Maksimalus mastelis	`MAX_ZOOM`	[skaičius: sveikas skaičius] Numatytas: 3	Didžiausias mastelis, kuriam teikiami kaladėlių duomenys. Nustatykite tarp 0 ir 24.
Apimtis Pasirinktinis	`EXTENT`	[apimtis] Numatytas: jokio	Maksimali braižomo žemėlapio ploto apimtis. Ribos turi apibrėžti visų mastelių dengiamą plotą.
Išvesties aplankas	`OUTPUT_DIRECTO`	[aplankas] Default: [Save to temporary folder]	Specification of the output vector tiles folder. *One of*: • Įrašyti į laikiną aplanką • Įrašyti į aplanką

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties aplankas	`OUTPUT_DIRECTO`	[aplankas]	A folder containing different subsets of the vector tiles files (`.pbf`) stored in subfolders corresponding to the zoom levels.

Pythono kodas

Algorithm ID: `native:writevectortiles_xyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.2 GDAL algorithm provider

GDAL (Geospatial Data Abstraction Library) is a translator library for raster and vector geospatial data formats. Algorithms in the Processing Framework are derived from the GDAL raster programs and GDAL vector programs.

24.2.1 Rastro analizė

Aspektas

Kuria krypties (aspekto) žemėlapį iš bet kurio GDAL palaikomo aukščio rastro. Aspektas yra nuolydžio paviršiaus kompaso kryptis. Pikseliai turės reikšmes 0-360°, matuojamas kampu nuo šiaurės, nurodančiu azimutą. Šiauriniame pusrutulyje šiaurinė nuolydžio pusė dažnai būna šešėlyje (žemas azimutas 0°-90°), o pietinė gauna daug saulės radiacijos (aukštesni azimutai 180°-270°).

Šis algoritmas kilęs iš GDAL DEM įrankio.

Numatytas meniu: *Rastras* ► *Analizė*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties aukščio rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	Juostos, kurią reikia naudoti kaip aukštį, numeris
Grąžinti trigonometrinį kampą vietoje azimuto	`TRIG_ANGLE`	[loginis] Numatytas: ne	Aktyvavus trigonometrinio kampo rezultatus gaunamos skirtingos kategorijos: 0° (Rytai), 90° (Šiaurė), 180° (Vakarai), 270° (Pietūs).

continues on next page

Table 24.847 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Plokštumo atveju grąžinti 0, o ne -9999	`ZERO_FLAT`	[loginis] Numatytas: ne	Įjungus šią parinktį, plotuose, kurie yra plokšti, bus įterpiama reikšmė 0 vietoje -9999.
Skaičiuoti kraštus	`COMPUTE_EDGES`	[loginis] Numatytas: ne	Pagal aukščio rastrą kuria kraštus
Naudoti Zevenbergen ir Thorne formulę vietoje Horno	`ZEVENBERGEN`	[loginis] Numatytas: ne	Aktyvuoja Zevenbergen ir Thorne formulę glotniems paviršiams
Aspektas	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro sluoksnis. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Aspektas	`OUTPUT`	[rastras]	Išvesties rastras su kampų reikšmėmis laipsniais

Pythono kodas

Alglritmo ID: `gdal:aspect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Spalvotas reljefas

Kuria spalvoto reljefo žemėlapį iš bet kokio GDAL-palaikomo aukščio rastro. Spalvotą reljefą galima naudoti aukščio vaizdavimui. Algoritmas sukuria 4 juostų rastrą su reikšmėmis, paskaičiuotomis iš aukščio ir tekstinio spalvų konfigūracijos failo. Pagal nutylėjimą spalvos tarp nurodytų aukščių suliejamos tolygiai, todėl gaunamas gražiai nuspalvintas aukščio rastras.

Šis algoritmas kilęs iš GDAL DEM įrankio.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties aukščio rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	Juostos, kurią reikia naudoti kaip aukštį, numeris
Skaičiuoti kraštus	`COMPUTE_EDGES`	[loginis] Numatytas: ne	Pagal aukščio rastrą kuria kraštus
Spalvų konfigūracijos failas	`COLOR_TABLE`	[file]	Tekstinis spalvų konfigūracijos failas
Atitikimo režimas	`MATCH_MODE`	[sąrašas] Numatyta: 2	Vienas iš: • 0 — Naudoti griežtą spalvos atitikimą • 1 — Naudoti artimiausią RGBA ketvertą • 2 — Naudoti tolygiai sulietas spalvas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

continues on next page

Table 24.850 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Spalvotas reljefas	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro sluoksnis. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Spalvotas reljefas	`OUTPUT`	[rastras]	4 juostų išvesties rastras

Pythono kodas

Algoritmo ID: `gdal:colorrelief`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Pildyti NėraDuomenų

Pildyti rastro regionus NėraDuomenų reikšmėmis interpoliuojant nuo kraštų. NėraDuomenų regionų reikšmės paskaičiuojamos pagal aplink esančias pikselių reikšmes naudojant atvirkštinio atstumo svorį. Po interpoliacijos vykdomas rezultatų glotninimas. Įvestis gali būti bet kuris GDAL palaikomas rastro sluoksnis. Šis algoritmas bendrai tinkamas interpoliuojant trūkstamus regionus daugmaž tolygiai kintančiuose rastruose (pavyzdžiui tokiuose kaip aukščio modeliai). Jis taipogi tinkamas pildant mažas skyles ir trūkius netolygiai kintančiuose vaizduose (kaip oro nuotraukose). Jis bendrai nelabai tinka interpoliuojant duomenis iš retai taškais išdėstytų duomenų.

Šis algoritmas kilęs iš GDAL fillnodata įrankio.

Numatytas meniu: *Rastras* ► *Analizė*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	Apdorojama juosta. NėraDuomenų reikšmės turi būti išreikštos reikšme 0.

continues on next page

Table 24.852 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Maksimalus atstumas (pikseliais) interpoliuojamų reikšmių paieškai	`DISTANCE`	[skaičius: sveikas skaičius] Numatytas: 10	Skaičius pikselių, kuriuo visomis kryptimis ieškoma reikšmių, iš kurių galima interpoliuoti
Glotninimo iteracijų skaičius, paleidžiamas po interpoliacijos	`ITERATIONS`	[skaičius: sveikas skaičius] Numatytas: 0	Skaičius 3x3 filtro iteracijų (0 ar daugiau) interpoliacijos rezultatų glotninimui.
Tinkamumo šablonas	`MASK_LAYER`	[rastras]	Rastro sluoksnis, kuris apibrėžia pildomus plotus.
Užpildytas	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	I6vesties rastro sluoksnio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Užpildytas	`OUTPUT`	[rastras]	Išvesties rastras

Pythono kodas

Algoritmo ID: `gdal:fillnodata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Grid (Data metrics)

Computes some data metrics using the specified window and output grid geometry.

This algorithm is derived from the GDAL grid utility.

Numatytas meniu: *Rastras* ► *Analizė*

Taip pat žiūrėkite

GDAL grid tutorial

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Taškų sluoksnis	`INPUT`	[vector: point]	Input point vector layer
Data metric to use	`METRIC`	[sąrašas] Numatytas: 0	Vienas iš: • 0 — Minimum, minimum value found in grid node search ellipse • 1 — Maximum, maximum value found in grid node search ellipse • 2 — Range, a difference between the minimum and maximum values found in grid node search ellipse • 3 — Count, a number of data points found in grid node search ellipse • 4 — Average distance, an average distance between the grid node (center of the search ellipse) and all of the data points found in grid node search ellipse • 5 — Average distance between points, an average distance between the data points found in grid node search ellipse. The distance between each pair of points within ellipse is calculated and average of all distances is set as a grid node value

continues on next page

Table 24.855 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
The first radius of search ellipse	`RADIUS_1`	[skaičius: slankaus kablelio] Numatytas: 0.0	The first radius (X axis if rotation angle is 0) of the search ellipse
The second radius of search ellipse	`RADIUS_2`	[skaičius: slankaus kablelio] Numatytas: 0.0	The second radius (Y axis if rotation angle is 0) of the search ellipse
Angle of search ellipse rotation in degrees (counter clockwise)	`ANGLE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Angle of ellipse rotation in degrees. Ellipse rotated counter clockwise.
Minimum number of data points to use	`MIN_POINTS`	[skaičius: slankaus kablelio] Numatytas: 0.0	Minimum number of data points to average. If less amount of points found the grid node considered empty and will be filled with NoData marker.
NoData	`NODATA`	[skaičius: slankaus kablelio] Numatytas: 0.0	NoData marker to fill empty points
Interpolated (data metrics)	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output raster layer with interpolated values. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Z value from field Pasirinktinis	`Z_FIELD`	[lentelės laukas: skaičius]	Field for the interpolation
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

continues on next page

Table 24.856 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas:	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 4 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Interpolated (data metrics)	`OUTPUT`	[rastras]	Output raster with interpolated values

Pythono kodas

Algorithm ID: `gdal:griddatametrics`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Grid (IDW with nearest neighbor searching)

Computes the Inverse Distance to a Power gridding combined to the nearest neighbor method. Ideal when a maximum number of data points to use is required.

This algorithm is derived from the GDAL grid utility.

Taip pat žiūrėkite

GDAL grid tutorial

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Taškų sluoksnis	`INPUT`	[vector: point]	Input point vector layer
Weighting power	`POWER`	[skaičius: slankaus kablelio] Numatyta: 2.0	Weighting power
Smoothing	`SMOOTHING`	[skaičius: slankaus kablelio] Numatytas: 0.0	Smoothing parameter
The radius of the search circle	`RADIUS`	[skaičius: slankaus kablelio] Numatytas: 1.0	The radius of the search circle
Maximum number of data points to use	`MAX_POINTS`	[skaičius: sveikas skaičius] Default: 12	Do not search for more points than this number.
Minimum number of data points to use	`MIN_POINTS`	[skaičius: sveikas skaičius] Numatytas: 0	Minimum number of data points to average. If less amount of points found the grid node considered empty and will be filled with NoData marker.
NoData	`NODATA`	[skaičius: slankaus kablelio] Numatytas: 0.0	NoData marker to fill empty points
Interpolated (IDW with NN search)	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output raster layer with interpolated values. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Z value from field Pasirinktinis	`Z_FIELD`	[lentelės laukas: skaičius]	Field for the interpolation

continues on next page

Table 24.859 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas:	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 4 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Interpolated (IDW with NN search)	`OUTPUT`	[rastras]	Output raster with interpolated values

Pythono kodas

Algorithm ID: `gdal:gridinversedistancenearestneighbor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Grid (Inverse distance to a power)

The Inverse Distance to a Power gridding method is a weighted average interpolator.

You should supply the input arrays with the scattered data values including coordinates of every data point and output grid geometry. The function will compute interpolated value for the given position in output grid.

This algorithm is derived from the GDAL grid utility.

Numatytas meniu: *Rastras* ► *Analizė*

Taip pat žiūrėkite

GDAL grid tutorial

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Taškų sluoksnis	`INPUT`	[vector: point]	Input point vector layer
Weighting power	`POWER`	[skaičius: slankaus kablelio] Numatyta: 2.0	Weighting power
Smothing	`SMOOTHING`	[skaičius: slankaus kablelio] Numatytas: 0.0	Smoothing parameter
The first radius of search ellipse	`RADIUS_1`	[skaičius: slankaus kablelio] Numatytas: 0.0	The first radius (X axis if rotation angle is 0) of the search ellipse
The second radius of search ellipse	`RADIUS_2`	[skaičius: slankaus kablelio] Numatytas: 0.0	The second radius (Y axis if rotation angle is 0) of the search ellipse
Angle of search ellipse rotation in degrees (counter clockwise)	`ANGLE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Angle of ellipse rotation in degrees. Ellipse rotated counter clockwise.
Maximum number of data points to use	`MAX_POINTS`	[skaičius: sveikas skaičius] Numatytas: 0	Do not search for more points than this number.

continues on next page

Table 24.861 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Minimum number of data points to use	`MIN_POINTS`	[skaičius: sveikas skaičius] Numatytas: 0	Minimum number of data points to average. If less amount of points found the grid node considered empty and will be filled with NoData marker.
NoData	`NODATA`	[skaičius: slankaus kablelio] Numatytas: 0.0	NoData marker to fill empty points
Interpolated (IDW)	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output raster layer with interpolated values. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Z value from field Pasirinktinis	`Z_FIELD`	[lentelės laukas: skaičius]	Field for the interpolation
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

continues on next page

Table 24.862 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas:	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 4 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Interpolated (IDW)	`OUTPUT`	[rastras]	Output raster with interpolated values

Pythono kodas

Algorithm ID: `gdal:gridinversedistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Grid (Linear)

The Linear method perform linear interpolation by computing a Delaunay triangulation of the point cloud, finding in which triangle of the triangulation the point is, and by doing linear interpolation from its barycentric coordinates within the triangle. If the point is not in any triangle, depending on the radius, the algorithm will use the value of the nearest point or the NoData value.

This algorithm is derived from the GDAL grid utility.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Taškų sluoksnis	`INPUT`	[vector: point]	Input point vector layer
Search distance	`RADIUS`	[skaičius: slankaus kablelio] Default: -1.0	In case the point to be interpolated does not fit into a triangle of the Delaunay triangulation, use that maximum distance to search a nearest neighbour, or use NoData otherwise. If set to `-1`, the search distance is infinite. If set to `0`, NoData value will be used.
NoData	`NODATA`	[skaičius: slankaus kablelio] Numatytas: 0.0	NoData marker to fill empty points
Interpolated (Linear)	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output raster layer with interpolated values. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Z value from field Pasirinktinis	`Z_FIELD`	[lentelės laukas: skaičius]	Field for the interpolation
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (`|`).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

continues on next page

Table 24.865 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas:	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 4 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Interpolated (Linear)	`OUTPUT`	[rastras]	Output raster with interpolated values

Pythono kodas

Algorithm ID: `gdal:gridlinear`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Grid (Moving average)

The Moving Average is a simple data averaging algorithm. It uses a moving window of elliptic form to search values and averages all data points within the window. Search ellipse can be rotated by specified angle, the center of ellipse located at the grid node. Also the minimum number of data points to average can be set, if there are not enough points in window, the grid node considered empty and will be filled with specified NoData value.

This algorithm is derived from the GDAL grid utility.

Numatytas meniu: *Rastras* ► *Analizė*

Taip pat žiūrėkite

GDAL grid tutorial

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Taškų sluoksnis	`INPUT`	[vector: point]	Input point vector layer
The first radius of search ellipse	`RADIUS_1`	[skaičius: slankaus kablelio] Numatytas: 0.0	The first radius (X axis if rotation angle is 0) of the search ellipse
The second radius of search ellipse	`RADIUS_2`	[skaičius: slankaus kablelio] Numatytas: 0.0	The second radius (Y axis if rotation angle is 0) of the search ellipse
Angle of search ellipse rotation in degrees (counter clockwise)	`ANGLE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Angle of ellipse rotation in degrees. Ellipse rotated counter clockwise.
Minimum number of data points to use	`MIN_POINTS`	[skaičius: sveikas skaičius] Numatytas: 0	Minimum number of data points to average. If less amount of points found the grid node considered empty and will be filled with NoData marker.
NoData	`NODATA`	[skaičius: slankaus kablelio] Numatytas: 0.0	NoData marker to fill empty points
Interpolated (moving average)	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Z value from field Pasirinktinis	`Z_FIELD`	[lentelės laukas: skaičius]	Field for the interpolation
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ,‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas:	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 4 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Interpolated (moving average)	`OUTPUT`	[rastras]	Output raster with interpolated values

Pythono kodas

Algorithm ID: `gdal:gridaverage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Grid (Nearest neighbor)

The Nearest Neighbor method doesn't perform any interpolation or smoothing, it just takes the value of nearest point found in grid node search ellipse and returns it as a result. If there are no points found, the specified NoData value will be returned.

This algorithm is derived from the GDAL grid utility.

Numatytas meniu: *Rastras* ► *Analizė*

Taip pat žiūrėkite

GDAL grid tutorial

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Taškų sluoksnis	`INPUT`	[vector: point]	Input point vector layer
The first radius of search ellipse	`RADIUS_1`	[skaičius: slankaus kablelio] Numatytas: 0.0	The first radius (X axis if rotation angle is 0) of the search ellipse
The second radius of search ellipse	`RADIUS_2`	[skaičius: slankaus kablelio] Numatytas: 0.0	The second radius (Y axis if rotation angle is 0) of the search ellipse
Angle of search ellipse rotation in degrees (counter clockwise)	`ANGLE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Angle of ellipse rotation in degrees. Ellipse rotated counter clockwise.
NoData	`NODATA`	[skaičius: slankaus kablelio] Numatytas: 0.0	NoData marker to fill empty points

continues on next page

Table 24.870 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Interpolated (Nearest neighbour)	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output raster layer with interpolated values. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Z value from field Pasirinktinis	`Z_FIELD`	[lentelės laukas: skaičius]	Field for the interpolation
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (`|`).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

continues on next page

Table 24.871 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas:	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 4 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Interpolated (Nearest neighbour)	`OUTPUT`	[rastras]	Output raster with interpolated values

Pythono kodas

Algorithm ID: `gdal:gridnearestneighbor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Hillshade

Outputs a raster with a nice shaded relief effect. It's very useful for visualizing the terrain. You can optionally specify the azimuth and altitude of the light source, a vertical exaggeration factor and a scaling factor to account for differences between vertical and horizontal units.

This algorithm is derived from the GDAL DEM utility .

Numatytas meniu: *Rastras ► Analizė*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Input Elevation raster layer
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	Band containing the elevation information
Z factor (vertical exaggeration)	`Z_FACTOR`	[skaičius: slankaus kablelio] Numatytas: 1.0	The factor exaggerates the height of the output elevation raster
Scale (ratio of vertical units to horizontal)	`SCALE`	[skaičius: slankaus kablelio] Numatytas: 1.0	The ratio of vertical units to horizontal units
Azimuth of the light	`AZIMUTH`	[skaičius: slankaus kablelio] Default: 315.0	Defines the azimuth of the light shining on the elevation raster in degrees. If it comes from the top of the raster the value is 0, if it comes from the east it is 90 a.s.o.
Altitude of the light	`ALTITUDE`	[skaičius: slankaus kablelio] Default: 45.0	Defines the altitude of the light, in degrees. 90 if the light comes from above the elevation raster, 0 if it is raking light.
Skaičiuoti kraštus	`COMPUTE_EDGES`	[loginis] Numatytas: ne	Pagal aukščio rastrą kuria kraštus
Use Zevenbergen&Thorı formula (instead of the Horn's one)	`ZEVENBERGEN`	[loginis] Numatytas: ne	Aktyvuoja Zevenbergen ir Thorne formulę glotniems paviršiams
Combined shading	`COMBINED`	[loginis] Numatytas: ne	
Multidirectional shading	`MULTIDIRECTION`	[loginis] Numatytas: ne	
Hillshade	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output raster layer with interpolated values. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Hillshade	`OUTPUT`	[rastras]	Output raster with interpolated values

Pythono kodas

Algorithm ID: `gdal:hillshade`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Near black

Converts nearly black/white borders to black.

This algorithm will scan an image and try to set all pixels that are nearly or exactly black, white or one or more custom colors around the collar to black or white. This is often used to „fix up“ lossy compressed airphotos so that color pixels can be treated as transparent when mosaicking.

This algorithm is derived from the GDAL nearblack utility.

Numatytas meniu: *Rastras* ► *Analizė*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Input Elevation raster layer
How far from black (white)	`NEAR`	[skaičius: sveikas skaičius] Default: 15	Select how far from black, white or custom colors the pixel values can be and still considered near black, white or custom color.
Search for nearly white pixels instead of nearly black	`WHITE`	[loginis] Numatytas: ne	Search for nearly white (255) pixels instead of nearly black pixels
Nearblack	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Nearblack	`OUTPUT`	[rastras]	Išvesties rastras

Pythono kodas

Algorithm ID: `gdal:nearblack`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Proximity (raster distance)

Generates a raster proximity map indicating the distance from the center of each pixel to the center of the nearest pixel identified as a target pixel. Target pixels are those in the source raster for which the raster pixel value is in the set of target pixel values.

This algorithm is derived from the GDAL proximity utility.

Numatytas meniu: *Rastras* ► *Analizė*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Input Elevation raster layer
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	Band containing the elevation information
List of target pixels Pasirinktinis	`VALUES`	[tekstas] Numatytas: ‚‘	Comma-separated list of pixel values in the source image to consider as target pixels. If not specified, all non-zero pixels will be considered target pixels.
Distance units	`UNITS`	[sąrašas] Numatytas: 1	Indicate whether distances generated should be in pixel or georeferenced coordinates. One of: • 0 — Georeferenced coordinates • 1 — Pixel coordinates
The maximum distance to generate Pasirinktinis	`MAX_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 0.0	The maximum distance to be generated. The NoData value will be used for pixels beyond this distance. Distance is interpreted according to the value of the *Distance units* parameter.
Value to apply to pixels within the maximum distance Pasirinktinis	`REPLACE`	[skaičius: slankaus kablelio] Numatytas: 0.0	Specify a value to apply to all pixels that are within the maximum distance from target pixels (including the target pixels) instead of a distance value.
NoData value to use for the destination proximity raster Pasirinktinis	`NODATA`	[skaičius: slankaus kablelio] Numatytas: 0.0	Specify the NoData value to use for the pixels beyond the maximum distance. If a NoData value is not provided, the output band will be queried for its NoData value. If the output band does not have a NoData value, then the value `65535` will be used.

continues on next page

Table 24.879 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Proximity map	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas:	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 4 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Proximity map	`OUTPUT`	[rastras]	Išvesties rastras

Pythono kodas

Algorithm ID: `gdal:proximity`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Roughness

Outputs a single-band raster with values computed from the elevation. Roughness is the degree of irregularity of the surface. It's calculated by the largest inter-cell difference of a central pixel and its surrounding cell. The determination of the roughness plays a role in the analysis of terrain elevation data, it's useful for calculations of the river morphology, in climatology and physical geography in general.

Šis algoritmas kilęs iš GDAL DEM įrankio.

Numatytas meniu: *Rastras* ► *Analizė*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties aukščio rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	Juostos, kurią reikia naudoti kaip aukštį, numeris
Skaičiuoti kraštus	`COMPUTE_EDGES`	[loginis] Numatytas: ne	Pagal aukščio rastrą kuria kraštus
Roughness	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Roughness	`OUTPUT`	[rastras]	Single-band output roughness raster. The value -9999 is used as NoData value.

Pythono kodas

Algorithm ID: `gdal:roughness`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Sieve

Removes raster polygons smaller than a provided threshold size (in pixels) and replaces them with the pixel value of the largest neighbour polygon. It is useful if you have a large amount of small areas on your raster map.

This algorithm is derived from the GDAL sieve utility.

Numatytas meniu: *Rastras* ► *Analizė*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties aukščio rastro sluoksnis
Threshold	`THRESHOLD`	[skaičius: sveikas skaičius] Numatytas: 10	Only raster polygons smaller than this size will be removed

continues on next page

Table 24.885 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Use 8-connectedness	EIGHT_CONNECTE	[loginis] Numatytas: ne	If False, consider as neighbours only the (4) pixels that connect the edges horizontally or vertically. If True, every pixel that touches an edge or a corner is considered as neighbour.
Do not use the default validity mask for the input band	NO_MASK	[loginis] Numatytas: ne	
Tinkamumo šablonas Pasirinktinis	MASK_LAYER	[rastras]	Validity mask to use instead of the default
Sieved	OUTPUT	[rastras] Numatytas: [Įrašyti į laikiną failą]	Specify the output raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomi komandinės eilutės parametrai Pasirinktinis	EXTRA	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sieved	OUTPUT	[rastras]	Output raster layer.

Pythono kodas

Algorithm ID: `gdal:sieve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Slope

Generates a slope map from any GDAL-supported elevation raster. Slope is the angle of inclination to the horizontal. You have the option of specifying the type of slope value you want: degrees or percent slope.

Šis algoritmas kilęs iš GDAL DEM įrankio.

Numatytas meniu: *Rastras* ► *Analizė*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Input Elevation raster layer
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	Band containing the elevation information
Ratio of vertical units to horizontal	`SCALE`	[skaičius: slankaus kablelio] Numatytas: 1.0	The ratio of vertical units to horizontal units
Express slope as percent (instead of degrees)	`AS_PERCENT`	[loginis] Numatytas: ne	Express slope as percent instead of degrees
Skaičiuoti kraštus	`COMPUTE_EDGES`	[loginis] Numatytas: ne	Pagal aukščio rastrą kuria kraštus
Use Zevenbergen&Thorı formula (instead of the Horn's one)	`ZEVENBERGEN`	[loginis] Numatytas: ne	Aktyvuoja Zevenbergen ir Thorne formulę glotniems paviršiams
Slope	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Slope	`OUTPUT`	[rastras]	Išvesties rastras

Pythono kodas

Algorithm ID: `gdal:slope`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Terrain Ruggedness Index (TRI)

Outputs a single-band raster with values computed from the elevation. TRI stands for Terrain Ruggedness Index, which is defined as the mean difference between a central pixel and its surrounding cells.

Šis algoritmas kilęs iš GDAL DEM įrankio.

Numatytas meniu: *Rastras* ► *Analizė*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties aukščio rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	Juostos, kurią reikia naudoti kaip aukštį, numeris
Skaičiuoti kraštus	`COMPUTE_EDGES`	[loginis] Numatytas: ne	Pagal aukščio rastrą kuria kraštus
Terrain Ruggedness Index	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Terrain Ruggedness Index	`OUTPUT`	[rastras]	Output ruggedness raster. The value -9999 is used as NoData value.

Pythono kodas

Algorithm ID: `gdal:triterrainruggednessindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Topographic Position Index (TPI)

Outputs a single-band raster with values computed from the elevation. TPI stands for Topographic Position Index, which is defined as the difference between a central pixel and the mean of its surrounding cells.

Šis algoritmas kilęs iš GDAL DEM įrankio.

Numatytas meniu: *Rastras* ► *Analizė*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties aukščio rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	The number of the band to use for elevation values

continues on next page

Table 24.894 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Skaičiuoti kraštus	`COMPUTE_EDGES`	[loginis] Numatytas: ne	Pagal aukščio rastrą kuria kraštus
Terrain Ruggedness Index	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Terrain Ruggedness Index	`OUTPUT`	[rastras]	Output raster.

Pythono kodas

Algorithm ID: `gdal:tpitopographicpositionindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.2.2 Rastro keitimas

gdal2xyz

Keičia rastro duomenis į XYZ ASCII failo formatą.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Keičiamas rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta] Numatytas: pirma įvesties sluoksnio juosta	Jei rastras turi kelias juostas, pasirinkite juostą, kurią norite keisti
Įvesties pikselio reikšmė, kurią reikia interpretuoti kaip NėraDuomenų Pasirinktinis	`NODATA_INPUT`	[skaičius: sveikas skaičius] Numatytas: jokio	Input pixel value to treat as „NoData“ **Dėmesio** Using this parameter requires QGIS installed with GDAL >= 3.7 (see *Help* ► *About* menu).
Paskirties NėraDuomenų Pasirinktinis	`NODATA_OUTPUT`	[skaičius: sveikas skaičius] Numatytas: jokio	Assign specified „NoData“ value to output. **Dėmesio** Using this parameter requires QGIS installed with GDAL >= 3.7 (see *Help* ► *About* menu).
Neišvesti NėraDuomenų reikšmių	`SKIP_NODATA`	[loginis] Numatytas: ne	Do not output „NoData“ values. **Dėmesio** Using this parameter requires QGIS installed with GDAL >= 3.3 (see *Help* ► *About* menu).
Kableliais atskirtos išvesties reikšmės	`CSV`	[loginis] Numatytas: ne	Nurodo, ar išvesties failas turi būti kableliais atskirtų reikšmių tipo (csv).
XYZ ASCII failas	`OUTPUT`	[file] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties failo specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
XYZ ASCII failas	`INPUT`	[lentelė]	Lentelės failas su reikšmėmis, eksportuojamomis iš rastro juostos.

Pythono kodas

Algoritmo ID: `gdal:gdal2xyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

PCT į RGB

Keičia 8 bitų paletės pietinį į 24 bitų RGB. Pakeis įvesties failo pseudo-spalvos juostą į norimo formatą RGB failą.

Šis algoritmas kilęs iš GDAL pct2rgb įrankio.

Numatytas meniu: *Rastras* ► *Keitimas*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties 8 bitų rastro pietinys
Juostos numeris	`BAND`	[rastro juosta] Numatytas: pirma įvesties sluoksnio juosta	Jei rastras turi kelias juostas, pasirinkite juostą, kurią norite keisti
Kurti RGBA failą	`RGBA`	[loginis] Numatytas: ne	Nurodo, ar išvesties failas turėtų būti RGBA tipo.
PCT į RGB	`OUTPUT`	[file] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties failo specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
PCT į RGB	`OUTPUT`	[rastras]	24 bitų RGB rastro piešinys

Pythono kodas

Algoritmo ID: `gdal:pcttorgb`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Polygonize (raster to vector)

Creates vector polygons for all connected regions of pixels in the raster sharing a common pixel value. Each polygon is created with an attribute indicating the pixel value of that polygon.

This algorithm is derived from the GDAL polygonize utility.

Numatytas meniu: *Rastras ► Keitimas*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta] Numatytas: pirma įvesties sluoksnio juosta	If the raster is multiband, choose the band you want to use
Name of the field to create	`FIELD`	[tekstas] Default: ‚DN‘	Specify the field name for the attributes of the connected regions.
Use 8-connectedness	`EIGHT_CONNECTE`	[loginis] Numatytas: ne	If not set, raster cells must have a common border to be considered connected (*4-connected*). If set, touching raster cells are also considered connected (*8-connected*).
Vectorized	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Įrašyti į laikiną failą]`	Specification of the output (polygon) vector layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Vectorized	`OUTPUT`	[vektorius: poligonas]	Output vector layer

Pythono kodas

Algorithm ID: `gdal:polygonize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Rearrange bands

Creates a new raster using selected band(s) from a given raster layer. The algorithm also makes it possible to reorder the bands for the newly-created raster.

Šis algoritmas kilęs iš GDAL translate įrankio.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Selected band(s)	`BANDS`	[raster band] [list] Numatytas: jokio	Ordered list of the bands to use to create the new raster
Pakeistas	`OUTPUT`	[rastras] Default: `[Save to temporary file]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ,‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas: 0	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Naudoti įvesties sluoksnio duomenų tipa • 1 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 2 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 3 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 4 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 5 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 6 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 7 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 8 — CInt16 (Kompleksinis Int16) • 9 — CInt32 (Kompleksinis Int32) • 10 — CFloat32 (Kompleksinis Float32) • 11 — CFloat64 (Kompleksinis Float64) • 12 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Pakeistas	`OUTPUT`	[rastras]	Output raster layer with rearranged bands.

Pythono kodas

Algorithm ID: `gdal:rearrange_bands`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

RGB to PCT

Converts a 24 bit RGB image into a 8 bit paletted. Computes an optimal pseudo-color table for the given RGB-image using a median cut algorithm on a downsampled RGB histogram. Then it converts the image into a pseudo-colored image using the color table. This conversion utilizes Floyd-Steinberg dithering (error diffusion) to maximize output image visual quality.

If you want to classify a raster map and want to reduce the number of classes it can be helpful to downsample your image with this algorithm before.

This algorithm is derived from the GDAL rgb2pct utility.

Numatytas meniu: *Rastras* ► *Keitimas*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Input (RGB) raster layer
Number of colors	`NCOLORS`	[skaičius: sveikas skaičius] Numatyta: 2	The number of colors the resulting image will contain. A value from 2-256 is possible.
RGB to PCT	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
RGB to PCT	`OUTPUT`	[rastras]	Output raster layer.

Pythono kodas

Algorithm ID: `gdal:rgbtopct`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Translate (convert format)

Converts raster data between different formats.

Šis algoritmas kilęs iš GDAL translate įrankio.

Numatytas meniu: *Rastras* ► *Keitimas*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Override the projection of the output file Pasirinktinis	`TARGET_CRS`	[crs]	Specify a projection for the output file
Priskirti nurodytą NėraDuomenų reikšmę išvesties juostoms Pasirinktinis	`NODATA`	[skaičius: slankaus kablelio] Numatytas: jokio	Defines the value to use for NoData in the output raster
Copy all subdatasets of this file to individual output files	`COPY_SUBDATASE`	[loginis] Numatytas: ne	Create individual files for subdatasets
Pakeistas	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specification of the output (translated) raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ,‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas: 0	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Naudoti įvesties sluoksnio duomenų tipa • 1 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 2 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 3 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 4 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 5 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 6 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 7 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 8 — CInt16 (Kompleksinis Int16) • 9 — CInt32 (Kompleksinis Int32) • 10 — CFloat32 (Kompleksinis Float32) • 11 — CFloat64 (Kompleksinis Float64) • 12 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Pakeistas	`OUTPUT`	[rastras]	Output (translated) raster layer.

Pythono kodas

Algorithm ID: `gdal:translate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.2.3 Rastro ištraukimas

Apkarpyti rastrą pagal apimtį

Apkarpo bet kokį GDAL palaikomą rastrą pagal pateiktą apimtį.

Šis algoritmas kilęs iš GDAL translate įrankio.

Numatytas meniu: *Rastras* ► *Ištraukimas*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties rastras

continues on next page

Table 24.912 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Apkarpymo apimtis	`EXTENT`	[apimtis]	Apimtis, kuri turi būti naudojama išvesties rastrui. Tik pikseliai, esantys nurodyto apibrėžiančio stačiakampio viduje, bus įtraukti į išvestį. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Permušti išvesties failo projekciją	`OVERCRS`	[loginis] Numatytas: ne	Įjungus šią parinktį, išvesties failui priskiriama įvesties failo CRS.
Priskirti nurodytą NėraDuomenų reikšmę išvesties juostoms Pasirinktinis	`NODATA`	[skaičius: slankaus kablelio] Numatytas: jokio	Nustato reikšmę, kuri turi būti įterpta išvesties rastro NėraDuomenų reikšmėms
Apkarpytas (apimtis)	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	I6vesties rastro sluoksnio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

continues on next page

Table 24.913 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties duomenų tipas	DATA_TYPE	[sąrašas] Numatytas: 0	Nustato išvesties rastro failo formatą. Parinktys: • 0 — Naudoti įvesties sluoksnio duomenų tipa • 1 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 2 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 3 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 4 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 5 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 6 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 7 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 8 — CInt16 (Kompleksinis Int16) • 9 — CInt32 (Kompleksinis Int32) • 10 — CFloat32 (Kompleksinis Float32) • 11 — CFloat64 (Kompleksinis Float64) • 12 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)
Papildomi komandinės eilutės parametrai Pasirinktinis	EXTRA	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Apkarpytas (apimtis)	OUTPUT	[rastras]	Išvesties rastro sluoksnis, apkarpytas pagal pateiktą apimtį

Pythono kodas

Algoritmo ID: `gdal:cliprasterbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Apkarpyti rastrą pagal šablono sluoksnį

Apkerpa bet kokį GDAL palaikomą rastrą pagal vektorinį šabloną.

Šis algoritmas kilęs iš GDAL warp įrankio.

Numatytas meniu: *Rastras* ► *Ištraukimas*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties rastras
Šablono sluoksnis	`MASK`	[vektorius: poligonas]	Vektorinis šablonas rastro apkirpimui
Šaltinio CRS	`SOURCE_CRS`	[crs]	Nustato įvesties rastrui naudojamą koordinačių atskaitos sistemą
Paskirties CRS	`TARGET_CRS`	[crs]	Nustato šablono sluoksniui naudojamą koordinačių atskaitos sistemą

continues on next page

Table 24.915 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Paskirties apimtis Pasirinktinis	`TARGET_EXTENT`	[apimtis]	Kuriamos išvesties failo apimtis Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Priskirti nurodytą NėraDuomenų reikšmę išvesties juostoms Pasirinktinis	`NODATA`	[skaičius: slankaus kablelio] Numatytas: jokio	Nustato reikšmę, kuri turi būti įterpta išvesties rastro NėraDuomenų reikšmėms
Kurti išvesties alfa juostą	`ALPHA_BAND`	[loginis] Numatytas: ne	Kuria rezultate alfa juostą. Alfa juostoje tada yra pikselių permatomumo reikšmės.
Atitikti apkirpto rastro apimtį pagal šablono sluoksnio apimtį	`CROP_TO_CUTLIN`	[loginis] Numatytas: Taip	Jei įjungta, pritaiko vektorinio sluoksnio apimtį išvesties rastrui.
Palikti įvesties rastro raišką	`KEEP_RESOLUTIO`	[loginis] Numatytas: ne	Išvesties rastro raiška nebus keičiama
Nustatyti išvesties failo raišką	`SET_RESOLUTION`	[loginis] Numatytas: ne	Ar turi būti nurodyta išvesties raiška (celės dydis)
Išvesties juostų X raiška Pasirinktinis	`X_RESOLUTION`	[skaičius: slankaus kablelio] Numatytas: jokio	Išvesties rastro celių plotis
Išvesties juostų Y raiška Pasirinktinis	`Y_RESOLUTION`	[skaičius: slankaus kablelio] Numatytas: jokio	Išvesties rastro celių aukštis
Naudoti kelių gijų lankstymo variantą	`MULTITHREADING`	[loginis] Numatytas: ne	Dvi gijos bus naudojamos piešinio gabalų apdorojimui ir įvesties/išvesties veiksmams vienu metu. Pastebėtina, kad pats skaičiavimas nėra dalinamas į gijas.

continues on next page

Table 24.915 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Apkarpytas (pagal šabloną)	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	I6vesties rastro sluoksnio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas: 0	Nustato išvesties rastro failo formatą. Parinktys: • 0 — Naudoti įvesties sluoksnio duomenų tipa • 1 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 2 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 3 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 4 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 5 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 6 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 7 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 8 — CInt16 (Kompleksinis Int16) • 9 — CInt32 (Kompleksinis Int32) • 10 — CFloat32 (Kompleksinis Float32) • 11 — CFloat64 (Kompleksinis Float64) • 12 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)

continues on next page

Table 24.916 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomi komandinės eilutės parametrai Pasirinktinis	EXTRA	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Apkarpytas (pagal šabloną)	OUTPUT	[rastras]	Išvesties rastro sluoksnis, apkarpytas pagal vektorinį sluoksnį

Pythono kodas

Algoritmo ID: `gdal:cliprasterbymasklayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Contour

Extracts contour lines from any GDAL-supported elevation raster.

This algorithm is derived from the GDAL contour utility.

Numatytas meniu: *Rastras* ► *Ištraukimas*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	INPUT	[rastras]	Input raster
Juostos numeris	BAND	[rastro juosta] Numatytas: 1	Raster band to create the contours from
Interval between contour lines	INTERVAL	[skaičius: slankaus kablelio] Numatytas: 10.0	Defines the interval between the contour lines in the given units of the elevation raster (minimum value 0)
Attribute name (if not set, no elevation attribute is attached) Pasirinktinis	FIELD_NAME	[tekstas] Default: ‚ELEV‘	Provides a name for the attribute in which to put the elevation.

continues on next page

Table 24.918 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Offset from zero relative to which to interpret intervals Pasirinktinis	`OFFSET`	[skaičius: slankaus kablelio] Numatytas: 0.0	
Contours	`OUTPUT`	[vektorius: linija] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties vektorinio sluoksnio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Produce 3D vector	`CREATE_3D`	[loginis] Numatytas: ne	Forces production of 3D vectors instead of 2D. Includes elevation at every vertex.
Treat all raster values as valid	`IGNORE_NODATA`	[loginis] Numatytas: ne	Ignores any NoData values in the dataset.
Įvesties pikselio reikšmė, kurią reikia interpretuoti kaip „NėraDuomenų“ Pasirinktinis	`NODATA`	[skaičius: slankaus kablelio] Numatytas: jokio	Nustato reikšmę, kuri turi būti įterpta išvesties rastro NėraDuomenų reikšmėms
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Add extra GDAL command line options. Refer to the corresponding GDAL utility documentation.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Contours	`OUTPUT`	[vektorius: linija]	Output vector layer with contour lines

Pythono kodas

Algorithm ID: `gdal:contour`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Contour Polygons

Extracts contour polygons from any GDAL-supported elevation raster.

This algorithm is derived from the GDAL contour utility.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Input raster
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	Raster band to create the contours from
Interval between contour lines	`INTERVAL`	[skaičius: slankaus kablelio] Numatytas: 10.0	Defines the interval between the contour lines in the given units of the elevation raster (minimum value 0)
Offset from zero relative to which to interpret intervals Pasirinktinis	`OFFSET`	[skaičius: slankaus kablelio] Numatytas: 0.0	
Attribute name for minimum elevation of contour polygon Pasirinktinis	`FIELD_NAME_MIN`	[tekstas] Default: ‚ELEV_MIN‘	Provides a name for the attribute in which to put the minimum elevation of contour polygon. If not provided no minimum elevation attribute is attached.
Attribute name for maximum elevation of contour polygon Pasirinktinis	`FIELD_NAME_MAX`	[tekstas] Default: ‚ELEV_MAX‘	Provides a name for the attribute in which to put the maximum elevation of contour polygon. If not provided no maximum elevation attribute is attached.
Contours	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties vektorinio sluoksnio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Produce 3D vector	`CREATE_3D`	[loginis] Numatytas: ne	Forces production of 3D vectors instead of 2D. Includes elevation at every vertex.
Treat all raster values as valid	`IGNORE_NODATA`	[loginis] Numatytas: ne	Ignores any NoData values in the dataset.
Įvesties pikselio reikšmė, kurią reikia interpretuoti kaip „NėraDuomenų“ Pasirinktinis	`NODATA`	[skaičius: slankaus kablelio] Numatytas: jokio	Nustato reikšmę, kuri turi būti įterpta išvesties rastro NėraDuomenų reikšmėms

continues on next page

Table 24.922 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomi komandinės eilutės parametrai Pasirinktinis	EXTRA	[tekstas] Numatytas: jokio	Add extra GDAL command line options. Refer to the corresponding GDAL utility documentation.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Contours	OUTPUT	[vektorius: poligonas]	Output vector layer with contour polygons

Pythono kodas

Algorithm ID: `gdal:contour_polygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.2.4 Rastro įvairūs

Kurti apžvalgas (piramides)

Norint pagreitinti braižymą, galima sukurti rastro sluoksnių apžvalgas (piramides). Apžvalgos yra žemesnės rezoliucijos duomenų kopijos, kurias QGIS naudoja priklausomai nuo mastelio.

Šis algoritmas kilęs iš GDAL addo įrankio.

Numatytas meniu: *Rastras ► Įvairūs*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	INPUT	[rastras]	Įvesties rastro sluoksnis
Išimti visas esamas apžvalgas	CLEAN	[boolean] Numatytas: ne	Išima iš rastro visas esamas apžvalgas. Pagal nutylėjimą jos nėra išimamos.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Apžvalgos lygiai	`LEVELS`	[tekstas] Numatyta: „2 4 8 16“	Nustato skaičių apžvalgos lygių, kurie skaičiuojami pagal pradinę įvesties rastro sluoksnio rezoliuciją. Pagal nutylėjimą vertinami 4 lygiai.
Perrinkimo metodas Pasirinktinis	`RESAMPLING`	[sąrašas] Numatytas: 0	Skaičiuoja apžvalgas naudojant nurodytą perrinkimo metodą. Galimi perrinkimo metodai yra: • 0 – Artimiausio kaimyno (`nearest`) • 1 – Vidurkio (`average`) • 2 – Gauso (`gauss`) • 3 – Kubinis (4x4 branduolio) (`cubic`) • 4 – Kubinis B-kreivės (4x4 branduolio) (`cubicspline`) • 5 – Lanczos (6x6 branduolio) (`lanczos`) • 6 – Vidutinio MP (`average_mp`) • 7 – Vidutinio Mag/Phase erdvėje (`average_magphase`) • 8 – Režimas (`mode`)
Apžvalgos formatas Pasirinktinis	`FORMAT`	[sąrašas] Numatytas: 0	Apžvalgas galima laikyti viduje arba išorėje kaip GTiff ar ERDAS piešinių failus. Pagal nutylėjimą apžvalgos laikomos išvesties rastre. Galimi formato metodai yra: • 0 – Vidinis (jei įmanoma) • 1 – Išorinis (GTiff .ovr) • 2 – Išorinis (ERDAS piešinys .aux)
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Piramidizuotas	`OUTPUT`	[rastras]	Išvesties rastro sluoksnis su apžvalgomis

Pythono kodas

Algoritmo ID: `gdal:overviews`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Kurti virtualų rastrą

Kuria VRT (virtualią duomenų aibę), kuri yra įvesties GDAL-palaikomų rastrų sąrašo mozaika. Naudodami mozaiką jūs galite sulieti kelis rastro failus.

Šis algoritmas kilęs iš GDAL buildvrt įrankio.

Numatytas meniu: *Rastras ► Įvairūs*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`INPUT`	[rastras] [sąrašas]	GDAL-palaikomi rastro sluoksniai.
Rezoliucija	`RESOLUTION`	[sąrašas] Numatytas: 0	Mozaikos išvesties rezoliucija. Pagal nutylėjimą bus parinkta vidutinė rastro failų rezoliucija. Parinktys: • 0 — Vidutinė (`average`) • 1 — Aukščiausia (`highest`) • 2 — Žemiausia (`lowest`)
Padėti kiekvieną įvesties failą į atskirą juostą	`SEPARATE`	[boolean] Numatytas: ne	Su „True“ jūs galite nurodyti, kad kiekvienas rastro failas eina į atskirą juostą VRT juostose.
Leisti projekcijų skirtumą	`PROJ_DIFFERENC`	[boolean] Numatytas: ne	Leidžia išvesties juostoms turėti skirtingas projekcijas, išskaičiuotas iš įvesties rastro sluoksnių projekcijų.
Virtualus	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro sluoksnio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Pridėti alfa slėpimo juostą į VRT, kai šaltinio rastras jokios neturi	`ADD_ALPHA`	[boolean] Numatytas: ne	Prideda alfa slėpimo juostą į VRT, kai šaltinio rastras jokios neturi.
Permušti išvesties failo projekciją Pasirinktinis	`ASSIGN_CRS`	[crs] Numatytas: jokio	Permuša išvesties failo projekciją. Perprojektavimas nedaromas.
Perrinkimo algoritmas	`RESAMPLING`	[sąrašas] Numatytas: 0	Naudojamas perrinkimo algoritmas. Parinktys: • 0 — Artimiausio kaimyno (`nearest`) • 1 — Bitiesinis (2x2 branduolio) (`bilinear`) • 2 — Kubinis (4x4 branduolio) (`cubic`) • 3 — Kubinis B-kreivės (4x4 branduolio) (`cubicspline`) • 4 — Lanczos (6x6 branduolio) (`lanczos`) • 5 — Vidutinis (`average`) • 6 — Režimas (`mode`)
NėraDuomenų reikšmė(s) įvesties juostoms (atskirtos tarpais) Pasirinktinis	`SRC_NODATA`	[tekstas] Numatytas: jokio	Tarpais atskirtos NėraDuomenų reikšmė(s) įvesties juostai(oms)
Papildomi komandinės eilutės parametrai	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Virtualus	`OUTPUT`	[rastras]	Išvesties rastro sluoksnis

Pythono kodas

Algoritmo ID: `gdal:buildvirtualraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

gdal2tiles

Kuria aplanką su mažomis kaladėlėmis ir metaduomenis pagal OSGeo Kaladėlių Žemėlapio Paslaugos (TMS) specifikaciją. Taipogi žiūrėkite OpenGIS Žiniatinklio žemėlapio kaladėlių paslaugos įgyvendinimo standartą. Taipogi sukuriami paprasti puslapiai su žiūryklėmis, paremtomis Google Maps, OpenLayers ir Leaflet. Norėdami tyrinėti jūsų žemėlapius internete naršyklėje, jums tik reikia įkelti sukurtą aplanką į serverį.

Šis algoritmas taipogi sukuria reikiamus metaduomenis, skirtus Google Earth (KML SuperOverlay), jei žemėlapis naudoja projekciją `EPSG:4326`.

ESRI pasaulio failai ir įtrauktas orientavimas naudojamas kaladėlių kūrimui, bet jūs galite publikuoti paveikslių ir be tinkamos orientacijos.

Šis algoritmas kilęs iš GDAL gdal2tiles įrankio.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	GDAL-palaikomas rastro sluoksnis.
Kaladėlių karpymo profilis	`PROFILE`	[sąrašas] Numatytas: 0	Vienas iš: • 0 — Merkatorius (`mercator`) • 1 — Geodezinis (`geodetic`) • 2 — Rastro (`raster`)
Braižomų mastelių skaičius Pasirinktinis	`ZOOM`	[tekstas] Numatytas: ‚‘	
Kurti žiniatinklio žiūryklę	`VIEWER`	[sąrašas] Numatytas: 0	Vienas iš: • 0 — Visas (`all`) • 1 — GoogleMaps (`google`) • 2 — OpenLayers (`openlayers`) • 3 — Leaflet (`leaflet`) • 4 — Jokios (`none`)
Žemėlapio pavadinimas Pasirinktinis	`TITLE`	[tekstas] Numatytas: ‚‘	
Žemėlapio autorinės teisės	`COPYRIGHT`	[tekstas] Numatytas: ‚‘	
Išvesties aplankas	`OUTPUT`	[aplankas] Numatytas: `[Įrašyti į laikiną aplanką]`	Įrašykite failų išvesties aplanką. *Vienas iš*: • Įrašyti į laikiną aplanką • Įrašyti į aplanką

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Perrinkimo metodas	`RESAMPLING`	[sąrašas] Numatytas: 0	Naudojamas perrinkimo algoritmas. Parinktys: • 0 — Vidutinė (`average`) • 1 — Artimiausio kaimyno (`near`) • 2 — Bitiesinis (2x2 branduolio) (`bilinear`) • 3 — Kubinis (4x4 branduolio) (`cubic`) • 4 — Kubinis B-kreivės (4x4 branduolio) (`cubicspline`) • 5 — Lanczos (6x6 branduolio) (`lanczos`) • 6 — Antialias (`antialias`)
Erdvinė atskaitos sistema, naudojama šaltinio įvesties duomenims Pasirinktinis	`SOURCE_CRS`	[crs] Numatytas: jokio	
Įvesties duomenims priskiriama permatomumo reikšmė Pasirinktinis	`NODATA`	[skaičius: slankaus kablelio] Numatytas: 0.0	
URL adresas, kur bus publikuojamos sukurtos kaladėlės Pasirinktinis	`URL`	[tekstas] Numatytas: ‚‘	
Google Maps API raktas (http://code.google.c Pasirinktinis	`GOOGLE_KEY`	[tekstas] Numatytas: ‚‘	Jūsų Google maps API raktas.
Bing Maps API raktas (https://www.bingm Pasirinktinis	`BING_KEY`	[tekstas] Numatytas: ‚‘	Jūsų Bing maps API raktas.
Kurti tik trūkstamus failus	`RESUME`	[boolean] Numatytas: ne	
Kurti Google Earth KML	`KML`	[boolean] Numatytas: ne	
Vengti automatinio KML kūrimo EPSG:4326	`NO_KML`	[boolean] Numatytas: ne	

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties aplankas	`OUTPUT`	[aplankas]	Išvesties aplankas (kaladėlėms)

Pythono kodas

Algoritmo ID: `gdal:gdal2tiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Sulieti

Sulieja rastro failus paprastu būdu. Čia galite naudoti pseudo-spalvų lentelę iš įvesties rastro ir apibrėžti išvesties rastro tipą. Visi piešiniai turi būti toje pačioje koordinačių sistemoje.

Šis algoritmas kilęs iš GDAL merge įrankio.

Numatytas meniu: *Rastras ► Įvairūs*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksniai	`INPUT`	[rastras] [sąrašas]	Įvesties rastro sluoksniai
Griebti pseudo-spalvų lentelę iš pirmo sluoksnio	`PCT`	[boolean] Numatytas: ne	Pseudo-spalvų lentelė iš pirmo sluoksnio bus naudojama spalvinimui
Padėti kiekvieną įvesties failą į atskirą juostą	`SEPARATE`	[boolean] Numatytas: ne	Padėti kiekvieną įvesties failą į atskirą juostą

continues on next page

Table 24.933 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas:	Apibrėžia išvesties rastro failo formatą. Parinktys: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 4 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)
Sulietas	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	I6vesties rastro sluoksnio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties pikselio reikšmė, kurią reikia interpretuoti kaip „NėraDuomenų“ Pasirinktinis	`NODATA_INPUT`	[skaičius: slankaus kablelio] Numatytas: jokio	Ignoruoja suliejamų failų pikselius su šia reikšme
Priskirti nurodytą „NėraDuomenų“ reikšmę išvesčiai Pasirinktinis	`NODATA_OUTPUT`	[skaičius: slankaus kablelio] Numatytas: jokio	Priskiria nurodytą NėraDuomenų reikšmę išvesties juostoms.

continues on next page

Table 24.934 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚'	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sulietas	`OUTPUT`	[rastras]	Išvesties rastro sluoksnis

Pythono kodas

Algoritmo ID: `gdal:merge`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Pansharpening

Performs a pan-sharpening operation. It can create a „classic“ output dataset (such as GeoTIFF), or a VRT dataset describing the pan-sharpening operation.

See GDAL Pansharpen.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Spectral dataset	`SPECTRAL`	[rastras]	Input (spectral) raster layer
Panchromatic dataset	`PANCHROMATIC`	[rastras]	Input (panchromatic) raster layer

continues on next page

Table 24.936 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvestis	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output (sharpened) raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Perrinkimo algoritmas	`RESAMPLING`	[sąrašas] Numatyta: 2	Naudojamas perrinkimo algoritmas. Parinktys: • 0 — Artimiausio kaimyno (`nearest`) • 1 — Bitiesinis (2x2 branduolio) (`bilinear`) • 2 — Kubinis (4x4 branduolio) (`cubic`) • 3 — Kubinis B-kreivės (4x4 branduolio) (`cubicspline`) • 4 — Lanczos (6x6 branduolio) (`lanczos`) • 5 — Vidutinis (`average`)
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (`|`).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvestis	`OUTPUT`	[rastras]	Output (sharpened) raster layer

Pythono kodas

Algorithm ID: `gdal:pansharp`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster calculator

Command line raster calculator with numpy syntax. Use any basic arithmetic supported by numpy arrays, such as +, -, *, and / along with logical operators, such as >. Note that all input rasters must have the same dimensions, but no projection checking is performed.

See the GDAL Raster Calculator utility docs.

Taip pat žiūrėkite

Raster calculator, *Raster calculator (virtual)*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input layer A	`INPUT_A`	[rastras]	First input raster layer (mandatory)
Number of raster band for A	`BAND_A`	[rastro juosta]	Band for input layer A (mandatory)
Input layer B Pasirinktinis	`INPUT_B`	[rastras] Numatytas: jokio	Second input raster layer
Number of raster band for B Pasirinktinis	`BAND_B`	[rastro juosta]	Band for input layer B
Input layer C Pasirinktinis	`INPUT_C`	[rastras] Numatytas: jokio	Third input raster layer
Number of raster band for C Pasirinktinis	`BAND_C`	[rastro juosta]	Band for input layer C
Input layer D Pasirinktinis	`INPUT_D`	[rastras] Numatytas: jokio	Fourth input raster layer
Number of raster band for D Pasirinktinis	`BAND_D`	[rastro juosta]	Band for input layer D
Input layer E Pasirinktinis	`INPUT_E`	[rastras] Numatytas: jokio	Fifth input raster layer
Number of raster band for E Pasirinktinis	`BAND_E`	[rastro juosta]	Band for input layer E
Input layer F Pasirinktinis	`INPUT_F`	[rastras]	Sixth input raster layer

continues on next page

Table 24.939 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Number of raster band for F Pasirinktinis	`BAND_F`	[rastro juosta] Numatytas: jokio	Band for input layer F
Calculation in gdalnumeric syntax using +-/* or any numpy array functions (i.e. logical_and())	`FORMULA`	[tekstas] Numatytas: ‚‘	The calculation formula. Examples: • `A*(A>0)` — outputs the value of the raster A if the value of A is greater than 0. If not, outputs 0. • `A*(A>0 and A>B)` — outputs the value of A if that value is bigger than 0 and bigger than the value of B. If not, outputs 0. • `A*logical_or(A<=177, A>=185)` — outputs the value of A if A <= 177 or A >= 185. If not, outputs 0. • `sqrt(A*A+B*B)` — Outputs the square root of the sum of the value of A squared and the value of B squared.
Set output NoData value Pasirinktinis	`NO_DATA`	[skaičius: slankaus kablelio] Numatytas: jokio	Value to use for NoData
Handling of extent differences	`EXTENT_OPT`	[sąrašas] Numatytas: 0	Determines how to handle rasters with different extents. Supported options are: • 0 — Ignore • 1 — Fail • 2 — Union • 3 — Intersect **Dėmesio** Using this parameter requires QGIS installed with GDAL >= 3.3 (see *Help* ► *About* menu).

continues on next page

Table 24.939 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Output extent Pasirinktinis	`INPUT`	[apimtis]	Custom extent of the output raster. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax` **Dėmesio** Using this parameter requires QGIS installed with GDAL >= 3.3 (see *Help* ► *About* menu).
Output raster type	`RTYPE`	[sąrašas] Numatytas:	Nurodo išvesties rastro failo duomenų tipą. Parinktys: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 4 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)

continues on next page

Table 24.939 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Calculated	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output (calculated) raster layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: ‚‘	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Calculated	`OUTPUT`	[rastras]	Output (calculated) raster layer

Pythono kodas

Algorithm ID: `gdal:rastercalculator`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Raster information

The gdalinfo program lists various information about a GDAL supported raster dataset.

This algorithm is derived from the GDAL info utility.

Numatytas meniu: *Rastras ► Įvairūs*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties rastro sluoksnis
Force computation of the actual min/max values for each band	`MIN_MAX`	[boolean] Numatytas: ne	Forces computation of the actual min/max values for each band in the dataset
Read and display image statistics (force computation if necessary)	`STATS`	[boolean] Numatytas: ne	Reads and displays image statistics. Forces computation if no statistics are stored in an image.
Suppress GCP info	`NO_GCP`	[boolean] Numatytas: ne	Suppresses ground control points list printing. It may be useful for datasets with huge amount of GCPs, such as L1B AVHRR or HDF4 MODIS which contain thousands of them.
Suppress metadata info	`NO_METADATA`	[boolean] Numatytas: ne	Suppresses metadata printing. Some datasets may contain a lot of metadata strings.
Sluoksnio informacija	`OUTPUT`	[html] Numatytas: `[Įrašyti į laikiną failą]`	Specify the HTML file for output. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sluoksnio informacija	`OUTPUT`	[html]	The HTML file containing information about the input raster layer

Pythono kodas

Algorithm ID: `gdal:gdalinfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Retile

Retiles a set of input tiles. All the input tiles must be georeferenced in the same coordinate system and have a matching number of bands. Optionally pyramid levels are generated.

This algorithm is derived from the GDAL Retile utility.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input files	`INPUT`	[rastras] [sąrašas]	The input raster files
Tile width	`TILE_SIZE_X`	[skaičius: sveikas skaičius] Default: 256	Width of the tiles in pixels (minimum 0)
Tile height	`TILE_SIZE_Y`	[skaičius: sveikas skaičius] Default: 256	Height of the tiles in pixels (minimum 0)
Overlap in pixels between consecutive tiles	`OVERLAP`	[skaičius: sveikas skaičius] Numatytas: 0	
Number of pyramid levels to build	`LEVELS`	[skaičius: sveikas skaičius] Numatytas: 1	Minimum: 0
Išvesties aplankas	`OUTPUT`	[aplankas] Numatytas: `[Įrašyti į laikiną aplanką]`	Įrašykite failų išvesties aplanką. *Vienas iš*: • Įrašyti į laikiną aplanką • Įrašyti į aplanką

continues on next page

Table 24.945 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
CSV file containing the tile(s) georeferencing information	`OUTPUT_CSV`	[file] Numatytas: `[Praleisti išvestį]`	Specify the output file for the tiles. *One of*: • Praleisti išvestį • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Source coordinate reference system Pasirinktinis	`SOURCE_CRS`	[crs] Numatytas: jokio	
Perrinkimo metodas	`RESAMPLING`	[sąrašas] Numatytas: 0	Naudojamas perrinkimo algoritmas. Parinktys: • 0 — Artimiausio kaimyno (`nearest`) • 1 — Bitiesinis (2x2 branduolio) (`bilinear`) • 2 — Kubinis (4x4 branduolio) (`cubic`) • 3 — Kubinis B-kreivės (4x4 branduolio) (`cubicspline`) • 4 — Lanczos (6x6 branduolio) (`lanczos`)
Column delimiter used in the CSV file Pasirinktinis	`DELIMITER`	[tekstas] Default: „;“	Delimiter to use in the CSV file containing the tile(s) georeferencing information
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (`|`).
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: ‚	Pridėkite papildomas GDAL komandinės eilutės parinktis

continues on next page

Table 24.946 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties duomenų tipas	DATA_TYPE	[sąrašas] Numatytas:	Apibrėžia išvesties rastro failo formatą. Parinktys: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 4 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)
Build only the pyramids	ONLY_PYRAMIDS	[boolean] Numatytas: ne	
Use separate directory for each tile row	DIR_FOR_ROW	[boolean] Numatytas: ne	

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties aplankas	OUTPUT	[aplankas]	The output folder for the tiles.
CSV file containing the tile(s) georeferencing information	OUTPUT_CSV	[file]	The CSV file with georeferencing information for the tiles.

Pythono kodas

Algorithm ID: `gdal:retile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Tile index

Builds a vector layer with a record for each input raster file, an attribute containing the filename, and a polygon geometry outlining the raster. This output is suitable for use with MapServer as a raster tileindex.

This algorithm is derived from the GDAL Tile Index utility.

Numatytas meniu: *Rastras ► Įvairūs*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Input files	`LAYERS`	[rastras] [sąrašas]	The input raster files. Can be multiple files.
Field name to hold the file path to the indexed rasters	`PATH_FIELD_NAM` Optional	[tekstas] Default: ‚location‘	The output field name to hold the file path/location to the indexed rasters.
Store absolute path to the indexed rasters	`ABSOLUTE_PATH`	[boolean] Numatytas: ne	Set whether the absolute path to the raster files is stored in the tile index file. By default the raster filenames will be put in the file exactly as they are specified in the command.
Skip files with different projection reference	`PROJ_DIFFERENC`	[boolean] Numatytas: ne	Only files with same projection as files already inserted in the tile index will be inserted. Default does not check projection and accepts all inputs.
Tile index	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Įrašyti į laikiną failą]`	Specify the polygon vector layer to write the index to. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Transform geometries to the given CRS Pasirinktinis	`TARGET_CRS`	[crs]	Geometries of input files will be transformed to the specified target coordinate reference system. Default creates simple rectangular polygons in the same coordinate reference system as the input rasters.
The name of the field to store the SRS of each tile Pasirinktinis	`CRS_FIELD_NAME`	[tekstas]	The name of the field to store the SRS of each tile
The format in which the CRS of each tile must be written	`CRS_FORMAT`	[sąrašas] Numatytas: 0	Format for the CRS. One of: • 0 – Auto (`AUTO`) • 1 – Well-known text (`WKT`) • 2 – EPSG (`EPSG`) • 3 – Proj.4 (`PROJ`)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Tile index	`OUTPUT`	[vektorius: poligonas]	The polygon vector layer with the tile index.

Pythono kodas

Algorithm ID: `gdal:tileindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Viewshed

Calculates a viewshed raster from an input raster DEM using method defined in Wang2000 for a user defined point.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Įvesties aukščio rastro sluoksnis
Juostos numeris	`BAND`	[rastro juosta] Numatytas: 1	Juostos, kurią reikia naudoti kaip aukštį, numeris
Observer location	`OBSERVER`	[coordinates]	The location of the observer
Observer height	`OBSERVER_HEIGH`	[skaičius: slankaus kablelio] Numatytas: 1.0	The altitude of the observer, in the DEM units
Target height	`TARGET_HEIGHT`	[skaičius: slankaus kablelio] Numatytas: 1.0	The altitude of the target element, in the DEM units
Maximum distance from observer to compute visibility	`MAX_DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 100.0	Maximum distance from observer to compute visibility, in the DEM units
Išvestis	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties rastro sluoksnis. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Papildomi komandinės eilutės parametrai	`EXTRA`	[tekstas] Numatytas: jokio	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvestis	`OUTPUT`	[rastras]	The raster layer displaying the viewshed.

Pythono kodas

Algorithm ID: `gdal:viewshed`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.2.5 Rastro projekcijos

Priskirti projekciją

Pritaiko koordinačių sistemą rastro duomenų rinkiniui.

Šis algoritmas kilęs iš GDAL edit įrankio.

Numatytas meniu: *Rastras* ► *Projekcijos*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT_LAYER`	[rastras]	Įvesties rastro sluoksnis
Norima CRS	`CRS`	[crs]	Išvesties sluoksnio projekcija (CRS)

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sluoksnis su projekcija	`OUTPUT`	[rastras]	Išvesties rastro sluoksnis (su nauja projekcijos informacija)

Pythono kodas

Algoritmo ID: `gdal:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Ištraukti projekciją

Ištraukia rastro failo projekciją ir įrašo ją į *pasaulio* failą su praplėtimu `.wld`.

Šis algoritmas kilęs iš GDAL srsinfo įrankio.

Numatytas meniu: *Rastras* ► *Projekcijos*

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties failas	`INPUT_LAYER`	[rastras]	Įvesties rastras. Rastro sluoksnis turi būti paremtas failu, nes algoritmas naudoja kelią iki rastro failo kaip vietą kuriamam failui `.wld`. Naudojant ne-failo rastro sluoksnį įvyks klaida.
Taipogi kurti .prj failą	`PRJ_FILE_CREAT`	[loginis] Numatytas: ne	Aktyvavus šią parinktį, taipogi kuriamas ir `.prj` failas, kuriame yra projekcijos informacija.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Pasaulio failas	`WORLD_FILE`	[failas]	Tekstinis failas su praplėtimu `.wld`, turinčiu rastro failo transformacijos parametrus.
ESRI Shapefile prj failas	`PRJ_FILE`	[failas]	Tekstinis failas su praplėtimu `.prj`, kuris aprašo CRS. Bus `None`, jei *Taipogi kurti .prj failą* nustatyta False.

Pythono kodas

Algoritmo ID: `gdal:extractprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Kraipymas (perprojektavimas)

Perprojektuoja rastro sluoksnį į kitą koordinačių atskaitos sistemą (CRS). Galima parinkti išvesties failo raišką ir perrinkimo metodą.

Šis algoritmas kilęs iš GDAL warp įrankio.

Numatytas meniu: *Rastras ► Projekcijos*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[rastras]	Perprojektuojamas įvesties rastras
Šaltinio CRS Pasirinktinis	`SOURCE_CRS`	[crs]	Apibrėžia įvesties rastro sluoksnio CRS
Paskirties CRS Pasirinktinis	`TARGET_CRS`	[crs] Numatytoji: `EPSG:4326`	Išvesties sluoksnio CRS
Naudojamas perrinkimo metodas	`RESAMPLING`	[sąrašas] Numatytas: 0	Naudojamas pikselių reikšmių perrinkimo metodas. Parinktys: • 0 — Artimiausias kaimynas • 1 — Bitiesinis (2x2 branduolio) • 2 — Kubinis (4x4 branduolio) • 3 — Kubinis B-kreivės (4x4 branduolio) • 4 — Lanczos (6x6 branduolio) • 5 — Vidurkis • 6 — Režimas • 7 — Maksimalus • 8 — Minimumas • 9 — Mediana • 10 — Pirmas kvartilis (Q1) • 11 — Trečias kvartilis (Q3)
NėraDuomenų reikšmė išvesties juostose Pasirinktinis	`NODATA`	[skaičius: slankaus kablelio] Numatytas: jokio	Nustato išvesties juostų NėraDuomenų reikšmę. Nenurodžius, NėraDuomenų reikšmės bus kopijuojamos iš šaltinio duomenų aibės.

continues on next page

Table 24.958 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties failo raiška paskirties geoorientuotais vienetais Pasirinktinis	`TARGET_RESOLUT`	[skaičius: slankaus kablelio] Numatytas: jokio	Apibrėžia perprojektavimo rezultato išvesties failo raišką
Perprojektuota	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	I6vesties rastro sluoksnio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).

continues on next page

Table 24.959 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas: 0	Apibrėžia išvesties rastro failo formatą. Parinktys: • 0 — Naudoti įvesties sluoksnio duomenų tipa • 1 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 2 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 3 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 4 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 5 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 6 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 7 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 8 — CInt16 (Kompleksinis Int16) • 9 — CInt32 (Kompleksinis Int32) • 10 — CFloat32 (Kompleksinis Float32) • 11 — CFloat64 (Kompleksinis Float64) • 12 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)
Kuriamo išvesties failo geoorientota apimtis Pasirinktinis	`TARGET_EXTENT`	[apimtis]	Nustato kuriamo išvesties failo geoorientotą apimtį (pagal nutylėjimą naudojant *Paskirties CRS*. Jei nurodyta, naudojant *paskirties rastro apimties CRS*). Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`

continues on next page

Table 24.959 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Paskirties rastro apimties CRS Pasirinktinis	`TARGET_EXTENT_`	[crs]	Nurodo CRS, kurioje reikia interpretuoti koordinates, nurodytas išvesties failo apimtyje. Nereikia šito painioti su išvesties duomenų aibės paskirties CRS. Vietoje to, tai yra patogus, pavyzdžiui kai išvesties koordinatės žinomos geografinėje ilgumos/platumos CRS, o rezultato norima projektuotoje koordinačių sistemoje.
Naudoti kelių gijų lankstymo variantą	`MULTITHREADING`	[loginis] Numatytas: ne	Dvi gijos bus naudojamos piešinio gabalų apdorojimui ir įvesties/išvesties veiksmams vienu metu. Pastebėtina, kad pats skaičiavimas nėra dalinamas į gijas.
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: jokio	Pridėti papildomas GDAL komandinės eilutės parinktis.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Perprojektuota	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	Perprojektuoti išvesties rastro sluoksnį

Pythono kodas

Algoritmo ID: `gdal:warpreproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.2.6 Vektorių keitimas

Keisti formatą

Keičia bet kokį OGC palaikomą vektorinį sluoksnį į kitą OGR palaikomą formatą.

Šis algoritmas kilęs iš ogr2ogr įrankio.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	Įvesties vektorinis sluoksnis
Keisti visus duomenų aibės sluoksnius	`CONVERT_ALL_LA`	[loginis] Numatytas: ne	Keičia visą duomenų aibę. Palaikomi šios parinkties išvesties formatai yra `GPKG` ir `GML`.
Pakeistas	`OUTPUT`	[toks pat, kaip įvestis]	Išvesties vektorinio sluoksnio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą… Parinkus `Įrašyti į failą` reikia nurodyti išvesties formatą. Palaikomi visi GDAL vektoriniai formatai. Parinkus `Įrašyti į laikiną failą` bus naudojamas QGIS numatytasis vektorinis formatas.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘ (jokių papildomų parinkčių)	Papildomos GDAL kūrimo parinktys.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Pakeistas	`OUTPUT`	[toks pat, kaip įvestis]	Išvesties vektorinis sluoksnis

Pythono kodas

Algoritmo ID: `gdal:convertformat`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Rasterizuoti (perrašyti su atributu)

Perrašo rastro failą vektorinio sluoksnio reikšmėmis. Naujos reikšmės priskiriamos pagal persidengiančio vektorinio geoobjekto atributo reikšmę.

Šis algoritmas kilęs iš GDAL rasterize įrankio.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Įvesties rastro sluoksnis	`INPUT_RASTER`	[rastras]	Įvesties rastro sluoksnis
Laukas, kurį naudoti perrašymo reikšmei Pasirinktinis	`FIELD`	[lentelės laukas: skaičius]	Nurodo atributo lauką, kurį naudoti nustatant pikselių reikšmes

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Pridėti reikšmes į esamo rastro reikšmes	`ADD`	[loginis] Numatytas: ne	Jei False, pikseliams priskiriama parinkto lauko reikšmė. Jei True, parinkto lauko reikšmė pridedama prie įvesties rastro sluoksnio reikšmės.
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: ‚‘	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Rasterizuotas	`OUTPUT`	[rastras]	Perrašytas įvesties rastro sluoksnis

Pythono kodas

Algoritmo ID: `gdal:rasterize_over`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Rasterizuoti (perrašyti konkrečia reikšme)

Perrašo dalį rastro sluoksnio nurodyta konkrečia reikšme. Perrašomi pikseliai parenkami pagal nurodyta (persidengiantį) vektorinį sluoksnį.

Šis algoritmas kilęs iš GDAL rasterize įrankio.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Įvesties rastro sluoksnis	`INPUT_RASTER`	[rastras]	Įvesties rastro sluoksnis
Konkreti išdeginama reikšmė	`BURN`	[skaičius: slankaus kablelio] Numatytas: 0.0	Išdeginama reikšmė

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Pridėti reikšmes į esamo rastro reikšmes	`ADD`	[loginis] Numatytas: ne	Jei False, pikseliams priskiriama konkreti reikšmė. Jei True, konkreti reikšmė pridedama prie įvesties rastro reikšmės.
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: ‚‘	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Rasterizuotas	`OUTPUT`	[rastras]	Perrašytas įvesties rastro sluoksnis

Pythono kodas

Algoritmo ID: `gdal:rasterize_over_fixed_value`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Rasterize (vector to raster)

Converts vector geometries (points, lines and polygons) into a raster image.

Šis algoritmas kilęs iš GDAL rasterize įrankio.

Numatytas meniu: *Rastras ► Keitimas*

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Laukas, kurį naudoti perrašymo reikšmei Pasirinktinis	`FIELD`	[lentelės laukas: skaičius]	Defines the attribute field from which the attributes for the pixels should be chosen
Konkreti išdeginama reikšmė Pasirinktinis	`BURN`	[skaičius: slankaus kablelio] Numatytas: 0.0	A fixed value to burn into a band for all features.
Burn value extracted from the „Z“ values of the feature	`USE_Z`	[loginis] Numatytas: ne	Indicates that a burn value should be extracted from the „Z“ values of the feature. Works with points and lines (linear interpolation along each segment). For polygons, works properly only if they are flat (same Z value for all vertices)
Output raster size units	`UNITS`	[sąrašas] Numatytas: 0	Units to use when defining the output raster size/resolution. One of: • 0 — Pixels • 1 — Georeferenced units

continues on next page

Table 24.970 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Width/Horizontal resolution	`WIDTH`	[skaičius: slankaus kablelio] Numatytas: 0.0	Sets the width (if size units is „Pixels“) or horizontal resolution (if size units is „Georeferenced units“) of the output raster. Minimum value: 0.0.
Height/Vertical resolution	`HEIGHT`	[skaičius: slankaus kablelio] Numatytas: 0.0	Sets the height (if size units is „Pixels“) or vertical resolution (if size units is „Georeferenced units“) of the output raster.
Output extent Pasirinktinis	`EXTENT`	[apimtis]	Extent of the output raster layer. If the extent is not specified, the minimum extent that covers the selected reference layer(s) will be used. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Priskirti nurodytą NėraDuomenų reikšmę išvesties juostoms Pasirinktinis	`NODATA`	[skaičius: slankaus kablelio] Numatytas: 0.0	Assigns a specified NoData value to output bands
Rasterizuotas	`OUTPUT`	[rastras] Numatytas: `[Įrašyti į laikiną failą]`	I6vesties rastro sluoksnio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą… For `Save to File`, the output format has to be specified. All GDAL raster formats are supported. For `Save to a Temporary File` the QGIS default raster format will be used.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘	Norint pridėti vieną ar daugiau kūrimo parinkčių, valdančių rastro kūrimą (spalvos, bloko dydis, failo suspaudimas…). Patogumui jūs galite remtis iš anksto sukurtais profiliais (žiūrėkite *GDAL tvarkyklės parinkčių skiltį*). Paketinis apdorojimas ir Modelių kūrimas: atskirkite kelias parinktis statmenu brūkšniu (|).
Išvesties duomenų tipas	`DATA_TYPE`	[sąrašas] Numatytas:	Nustato išvesties rastro failo formatą. Parinktys: • 0 — Byte (aštuonių bitų sveikas skaičius be ženklo (quint8)) • 1 — Int16 (Šešiolikos bitų sveikas skaičius su ženklu (qint16)) • 2 — UInt16 (Šešiolikos bitų sveikas skaičius be ženklo (quint16)) • 3 — UInt32 (Trisdešimt dviejų bitų sveikas skaičius be ženklo (quint32)) • 4 — Int32 (Trisdešimt dviejų bitų sveikas skaičius su ženklu (qint32)) • 5 — Float32 (Trisdešimt dviejų bitų slankaus kablelio skaičius (float)) • 6 — Float64 (Šešiasdešimt keturių bitų slankaus kablelio skaičius (double)) • 7 — CInt16 (Kompleksinis Int16) • 8 — CInt32 (Kompleksinis Int32) • 9 — CFloat32 (Kompleksinis Float32) • 10 — CFloat64 (Kompleksinis Float64) • 11 — Int8 (Aštuonių bitų sveikas skaičius su ženklu (qint8)) Galimi variantai priklauso nuo GDAL versijos, su kuria sukurtas QGIS (žiūrėkite meniu *Pagalba* ► *Apie*)
Pre-initialize the output image with value Pasirinktinis	`INIT`	[skaičius: slankaus kablelio]	Pre-initializes the output image bands with this value. Not marked as the NoData value in the output file. The same value is used in all the bands.
Invert rasterization	`INVERT`	[loginis] Numatytas: ne	Burns the fixed burn value, or the burn value associated with the first feature into all parts of the image not inside the provided polygon.
Papildomi komandinės eilutės parametrai Pasirinktinis	`EXTRA`	[tekstas] Numatytas: ‚‘	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Rasterizuotas	`OUTPUT`	[rastras]	Išvesties rastro sluoksnis

Pythono kodas

Algorithm ID: `gdal:rasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.2.7 Vektorių geoapdorojimas

Vektorių buferiai

Kurti buferius aplink vektorinių sluoksnių geoobjektus.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Geometrijos stulpelio pavadinimas	`GEOMETRY`	[tekstas] Numatytas: ‚geometry‘	Įvesties sluoksnio geometrijos stulpelio pavadinimas
Buferio atstumas	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 10.0	Minimalus: 0.0
Sulieti pagal atributus Pasirinktinis	`FIELD`	[lentelės laukas: bet koks] Numatytas: jokio	Suliejimui naudojamas laukas
Sulieti rezultatus	`DISSOLVE`	[loginis] Numatytas: ne	Nustačius rezultatas bus sulietas. Jei nenurodytas joks suliejimo laukas, visi buferiai suliejami į vieną geoobjektą.
Kurti vieną geoobjektą kiekvienai geometrijai bet kokio tipo geometrijų rinkiniui pradiniame faile	`EXPLODE_COLLEC`	[loginis] Numatytas: ne	

continues on next page

Table 24.973 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Buferis	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite išvesties buferio sluoksnį. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘ (jokių papildomų parinkčių)	Papildomos GDAL kūrimo parinktys.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Buferis	`OUTPUT`	[vektorius: poligonas]	Išvesties buferių sluoksnis

Pythono kodas

Algoritmo ID: `gdal:buffervectors`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Apkirpti vektorių pagal apimtį

Apkerpa bet kokį OGR palaikomą vektorinį failą pagal pateiktą apimtį.

Šis algoritmas kilęs iš GDAL ogr2ogr įrankio.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Apkirpimo apimtis	`EXTENT`	[apimtis]	Apibrėžia apimties stačiakampį, kuris turi būti naudojamas išvesties vektoriniam failui. Jis turi būti apibrėžtas naudojant paskirties CRS koordinates. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Apkarpytas (apimtis)	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite išvesties (apkarpytą) sluoksnį. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘ (jokių papildomų parinkčių)	Papildomos GDAL kūrimo parinktys.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Apkarpytas (apimtis)	`OUTPUT`	[toks pat, kaip įvestis]	Išvesties (apkarpytas) sluoksnis. Numatytasis formatas yra „ESRI Shapefile“.

Pythono kodas

Algoritmo ID: `gdal:clipvectorbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Apkarpyti vektorių pagal šablono sluoksnį

Apkerpa bet kokį OGR-palaikomą vektorinį sluoksnį pagal šablono poligonų sluoksnį.

Šis algoritmas kilęs iš GDAL ogr2ogr įrankio.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Šablono sluoksnis	`MASK`	[vektorius: poligonas]	Sluoksnis, kuris turi būti naudojamas kaip apkirpimo apimtis įvesties vektoriniam sluoksniui.
Apkarpytas (pagal šabloną)	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Išvesties (paslėptas) sluoksnis. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘ (jokių papildomų parinkčių)	Papildomos GDAL kūrimo parinktys.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Apkarpytas (pagal šabloną)	`OUTPUT`	[toks pat, kaip įvestis]	Išvesties (apkarpytas) sluoksnis. Numatytasis formatas yra „ESRI Shapefile“.

Pythono kodas

Algoritmo ID: `gdal:clipvectorbypolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Dissolve

Dissolve (combine) geometries that have the same value for a given attribute / field. The output geometries are multipart.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	The input layer to dissolve
Dissolve field Pasirinktinis	`FIELD`	[lentelės laukas: bet koks]	The field of the input layer to use for dissolving
Geometrijos stulpelio pavadinimas	`GEOMETRY`	[tekstas] Numatytas: ‚geometry‘	The name of the input layer geometry column to use for dissolving.

continues on next page

Table 24.982 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Sulietas	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite išvesties sluoksnį. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Kurti vieną geoobjektą kiekvienai geometrijai bet kokio tipo geometrijų rinkiniui pradiniame faile	`EXPLODE_COLLEC`	[loginis] Numatytas: ne	Produce one feature for each geometry in any kind of geometry collection in the source file
Keep input attributes	`KEEP_ATTRIBUTE`	[loginis] Numatytas: ne	Keep all attributes from the input layer
Count dissolved features	`COUNT_FEATURES`	[loginis] Numatytas: ne	Count the dissolved features and include it in the output layer.
Compute area and perimeter of dissolved features	`COMPUTE_AREA`	[loginis] Numatytas: ne	Compute the area and perimeter of dissolved features and include them in the output layer
Compute min/max/sum/mean for attribute	`COMPUTE_STATIS`	[loginis] Numatytas: ne	Calculate statistics (min, max, sum and mean) for the numeric attribute specified and include them in the output layer
Numeric attribute to calculate statistics on Pasirinktinis	`STATISTICS_ATT`	[lentelės laukas: skaičius]	The numeric attribute to calculate statistics on
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚ (jokių papildomų parinkčių)	Papildomos GDAL kūrimo parinktys.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sulietas	`OUTPUT`	[toks pat, kaip įvestis]	The output multipart geometry layer (with dissolved geometries)

Pythono kodas

Algorithm ID: `gdal:dissolve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Offset curve

Offsets lines by a specified distance. Positive distances will offset lines to the left, and negative distances will offset them to the right.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	The input line layer
Geometrijos stulpelio pavadinimas	`GEOMETRY`	[tekstas] Numatytas: ‚geometry‘	Įvesties sluoksnio geometrijos stulpelio pavadinimas
Offset distance (left-sided: positive, right-sided: negative)	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 10.0	
Offset curve	`OUTPUT`	[vektorius: linija] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output line layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘ (jokių papildomų parinkčių)	Papildomos GDAL kūrimo parinktys.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Offset curve	`OUTPUT`	[vektorius: linija]	The output offset curve layer

Pythono kodas

Algorithm ID: `gdal:offsetcurve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

One side buffer

Creates a buffer on one side (right or left) of the lines in a line vector layer.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	The input line layer
Geometrijos stulpelio pavadinimas	`GEOMETRY`	[tekstas] Numatytas: ‚geometry'	Įvesties sluoksnio geometrijos stulpelio pavadinimas
Buferio atstumas	`DISTANCE`	[skaičius: slankaus kablelio] Numatytas: 10.0	
Buffer side	`BUFFER_SIDE`	[sąrašas] Numatytas: 0	Vienas iš: • 0 — Right • 1 — Left
Sulieti pagal atributus Pasirinktinis	`FIELD`	[lentelės laukas: bet koks] Numatytas: jokio	Suliejimui naudojamas laukas
Dissolve all results	`DISSOLVE`	[loginis] Numatytas: ne	**If set, the result is dissolved.** If no field is set for dissolving, all the buffers are dissolved into one feature.

continues on next page

Table 24.988 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Kurti vieną geoobjektą kiekvienai geometrijai bet kokio tipo geometrijų rinkiniui pradiniame faile	`EXPLODE_COLLEC`	[loginis] Numatytas: ne	
One-sided buffer	`OUTPUT`	[vektorius: poligonas] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite išvesties buferio sluoksnį. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘ (jokių papildomų parinkčių)	Papildomos GDAL kūrimo parinktys.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
One-sided buffer	`OUTPUT`	[vektorius: poligonas]	Išvesties buferių sluoksnis

Pythono kodas

Algorithm ID: `gdal:onesidebuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Points along lines

Generates a point on each line of a line vector layer at a distance from start. The distance is provided as a fraction of the line length.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: linija]	The input line layer
Geometrijos stulpelio pavadinimas	`GEOMETRY`	[tekstas] Numatytas: ‚geometry'	Įvesties sluoksnio geometrijos stulpelio pavadinimas
Distance from line start represented as a fraction of line length	`DISTANCE`	[skaičius: slankaus kablelio] Numatyta: 0.5	
Points along lines	`OUTPUT`	[vector: point] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output point layer. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚' (jokių papildomų parinkčių)	Papildomos GDAL kūrimo parinktys.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Taškai palei liniją	`OUTPUT`	[vector: point]	The output point layer

Pythono kodas

Algorithm ID: `gdal:pointsalonglines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

24.2.8 Įvairūs vektoriniai

Kurti virtualų vektorių

Kuria virtualų vektorinį sluoksnį, kuriame yra vektorinių sluoksnių aibė.

Šis algoritmas ypatingai naudingas, kai kitam algoritmui reikia kelių sluoksnių, bet jis priima tik vieną `vrt`, kuriame nurodyti sluoksniai.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties duomenų šaltiniai	`INPUT`	[vektorius: bet koks] [sąrašas]	Parinkite vektorinius sluoksnius, kuriuos norite naudoti kuriant virtualų vektorių
Kurti „sąjungos“ VRT	`UNIONED`	[loginis] Numatytas: ne	Įjunkite, jei norite apjungti visus vektorius į vieną `vrt` failą
Virtualus vektorius	`OUTPUT`	[toks pat, kaip įvestis] Numatytas: `[Įrašyti į laikiną failą]`	Nurodykite išvesties sluoksnį, kuriame bus tik dublikatai. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą…

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Virtualus vektorius	`OUTPUT`	[vektorius: bet koks]	Išvesties vektorinis sluoksnis, sukurtas iš parinktų šaltinių

Pythono kodas

Algoritmo ID: `gdal:buildvirtualvector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Vykdyti SQL

Vykdo šaltinio sluoksnyje paprastą arba sudėtingą SQL sintaksės užklausą. Užklausos rezultatas pridedamas kaip naujas sluoksnis.

Šis algoritmas kilęs iš GDAL ogr2ogr įrankio.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	OGR-palaikomas įvesties vektorinis sluoksnis
SQL išraiška	`SQL`	[tekstas]	Apibrėžia SQL užklausą, pavyzdžiui `SELECT * FROM mano_lentele WHERE pavadinimas is not null`.
SQL dialektas	`DIALECT`	[sąrašas] Numatytas: 0	Naudojamas SQL dialektas. Vienas iš: • 0 — Jokio • 1 — OGR SQL • 2 — SQLite
SQL rezultatas	`OUTPUT`	[vektorius: bet koks]	Išvesties sluoksnio specifikacija. *Vienas iš*: • Įrašyti į laikiną failą • Įrašyti į failą… Parinkus `Įrašyti į failą` reikia nurodyti išvesties formatą. Palaikomi visi GDAL vektoriniai formatai. Parinkus `Įrašyti į laikiną failą`, bus naudojamas numatytasis vektorinių išvesties sluoksnių formatas.

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Papildomos kūrimo parinktys Pasirinktinis	`CREATION_OPTIO` (for QGIS <= 3.42, this was `OPTIONS`)	[tekstas] Numatytas: ‚‘ (jokių papildomų parinkčių)	Papildomos GDAL kūrimo parinktys.

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
SQL rezultatas	`OUTPUT`	[vektorius: bet koks]	Pagal užklausą sukurtas vektorinis sluoksnis

Pythono kodas

Algoritmo ID: `gdal:executesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export to PostgreSQL (available connections)

Imports vector layers inside a PostgreSQL database on the basis of an available connection. The connection has to *be defined properly* beforehand. Be aware that the checkboxes ‚Save Username' and ‚Save Password' are activated. Then you can use the algorithm.

Šis algoritmas kilęs iš GDAL ogr2ogr įrankio.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Duomenų bazė (jungties pavadinimas)	`DATABASE`	[tekstas]	The PostgreSQL database to connect to
Įvesties sluoksnis	`INPUT`	[vektorius: bet koks]	OGR-supported vector layer to export to the database
Shape encoding Pasirinktinis	`SHAPE_ENCODING`	[tekstas] Numatytas: ‚'	Sets the encoding to apply to the data
Output geometry type	`GTYPE`	[sąrašas] Numatytas: 0	Defines the output geometry type. One of: • 0 — • 1 — NONE • 2 — GEOMETRY • 3 — POINT • 4 — LINESTRING • 5 — POLYGON • 6 — GEOMETRYCOLLECTION • 7 — MULTIPOINT • 8 — MULTIPOLYGON • 9 — MULTILINESTRING • 10 — CIRCULARSTRING • 11 — COUMPOUNDCURVE • 12 — CURVEPOLYGON • 13 — MULTICURVE • 14 — MULTISURFACE • 15 — CONVERT_TO_LINEAR • 16 — CONVERT_TO_CURVE
Assign an output CRS Pasirinktinis	`A_SRS`	[crs] Numatytas: jokio	Defines the output CRS of the database table
Reproject to this CRS on output Pasirinktinis	`T_SRS`	[crs] Numatytas: jokio	Reprojects/transforms to this CRS on output

continues on next page

Table 24.999 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Override source CRS Pasirinktinis	S_SRS	[crs] Numatytas: jokio	Overrides the input layer CRS
Schema (schemos pavadinimas) Pasirinktinis	SCHEMA	[tekstas] Numatyta: ‚public'	Defines the schema for the database table
Table to export to (leave blank to use layer name) Pasirinktinis	TABLE	[tekstas] Numatytas: ‚'	Defines a name for the table that will be imported into the database. By default the table name is the name of the input vector file.
Primary Key (new field) Pasirinktinis	PK	[tekstas] Default: ‚id'	Defines which attribute field will be the primary key of the database table
Primary Key (existing field, used if the above option is left empty) Pasirinktinis	PRIMARY_KEY	[tablefield: any] Numatytas: jokio	Defines which attribute field in the exported layer will be the primary key of the database table
Geometrijos stulpelio pavadinimas Pasirinktinis	GEOCOLUMN	[tekstas] Numatytas: ‚geom'	Defines in which attribute field of the database there will be the geometry information
Vector dimensions Pasirinktinis	DIM	[sąrašas] Default: 0 (2D)	Defines the coordinate dimensions of the imported vector data. One of: • 0 — 2 (XY) • 1 — 3 (XYZ) • 2 — 4 (XYZM)
Distance tolerance for simplification Pasirinktinis	SIMPLIFY	[tekstas] Numatytas: ‚'	Defines a distance tolerance for the simplification of the vector geometries to be imported. By default there is no simplification.
Maximum distance between 2 nodes (densification) Pasirinktinis	SEGMENTIZE	[tekstas] Numatytas: ‚'	The maximum distance between two nodes. Used to create intermediate points. By default there is no densification.

continues on next page

Table 24.999 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Select features by extent (defined in input layer CRS) Pasirinktinis	SPAT	[apimtis] Numatytas: jokio	You can select features from a given extent that will be in the output table. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Clip the input layer using the above (rectangle) extent	CLIP	[loginis] Numatytas: ne	The input layer will be clipped by the extent you defined before
Select features using a SQL „WHERE“ statement (Ex: column=“value“) Pasirinktinis	WHERE	[tekstas] Numatytas: ‚	Defines with a SQL „WHERE“ statement which features should be selected from the input layer
Group N features per transaction (Default: 2000) Pasirinktinis	GT	[tekstas] Numatytas: ‚	You can group the input features in transactions where N defines the size. By default N limits the transaction size to 20000 features.
Overwrite existing table	OVERWRITE	[loginis] Numatytas: Taip	If there is a table with the same name in the database, and if this option is set to True, the table will be overwritten.
Append to existing table	APPEND	[loginis] Numatytas: ne	If checked / True the vector data will be appended to an existing table. New fields found in the input layer are ignored. By default a new table will be created.
Append and add new fields to existing table	ADDFIELDS	[loginis] Numatytas: ne	If activated the vector data will be appended to an existing table, there won’t be a new table created. New fields found in input layer are added to the table. By default a new table will be created.
Do not launder columns/table names	LAUNDER	[loginis] Numatytas: ne	With this option checked you can prevent the default behaviour (converting column names to lowercase, removing spaces and other invalid characters).
Do not create Spatial Index	INDEX	[loginis] Numatytas: ne	Prevents a spatial index for the output table from being created. By default, a spatial index is added.
Continue after a failure, skipping the failed feature	SKIPFAILURES	[loginis] Numatytas: ne	Continue after a failure, skipping the failed feature.

continues on next page

Table 24.999 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Validate geometries based on Simple Features specification	MAKEVALID	[loginis] Numatytas: ne	Applies cleanup operations on geometries to ensure they are valid regarding the rules of the Simple Features specification.
Promote to Multipart	PROMOTETOMULTI	[loginis] Numatytas: Taip	Casts features geometry type to multipart in the output table
Keep width and precision of input attributes	PRECISION	[loginis] Numatytas: Taip	Avoids modifying column attributes to comply with input data
Papildomos kūrimo parinktys Pasirinktinis	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[tekstas] Numatytas: ‚‘ (jokių papildomų parinkčių)	Papildomos GDAL kūrimo parinktys.

Išvestys

This algorithm has no output.

Pythono kodas

Algorithm ID: `gdal:importvectorintopostgisdatabaseavailableconnections`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Export to PostgreSQL (new connection)

Imports vector layers inside a PostgreSQL database. A new connection to the PostgreSQL database must be created.

Šis algoritmas kilęs iš GDAL ogr2ogr įrankio.

Parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	INPUT	[vektorius: bet koks]	OGR-supported vector layer to export to the database
Shape encoding Pasirinktinis	SHAPE_ENCODING	[tekstas] Numatytas: ‚‘	Sets the encoding to apply to the data

continues on next page

Table 24.1000 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Output geometry type	GTYPE	[sąrašas] Numatytas: 0	Defines the output geometry type. One of: • 0 — • 1 — NONE • 2 — GEOMETRY • 3 — POINT • 4 — LINESTRING • 5 — POLYGON • 6 — GEOMETRYCOLLECTION • 7 — MULTIPOINT • 8 — MULTIPOLYGON • 9 — MULTILINESTRING • 10 — CIRCULARSTRING • 11 — COUMPOUNDCURVE • 12 — CURVEPOLYGON • 13 — MULTICURVE • 14 — MULTISURFACE • 15 — CONVERT_TO_LINEAR • 16 — CONVERT_TO_CURVE
Assign an output CRS Pasirinktinis	A_SRS	[crs] Numatytas: jokio	Defines the output CRS of the database table
Reproject to this CRS on output Pasirinktinis	T_SRS	[crs] Numatytas: jokio	Reprojects/transforms to this CRS on output
Override source CRS Pasirinktinis	S_SRS	[crs] Numatytas: jokio	Overrides the input layer CRS
Host Pasirinktinis	HOST	[tekstas] Default: ‚localhost‘	Name of the database host
Port Pasirinktinis	PORT	[tekstas] Default: ‚5432‘	Port number the PostgreSQL database server listens on
Username Pasirinktinis	USER	[tekstas] Numatytas: ‚‘	User name used to log in to the database
Database name Pasirinktinis	DBNAME	[tekstas] Numatytas: ‚‘	Name of the database
Password Pasirinktinis	PASSWORD	[tekstas] Numatytas: ‚‘	Password used with Username to connect to the database
Schema (schemos pavadinimas) Pasirinktinis	SCHEMA	[tekstas] Numatyta: ‚public‘	Defines the schema for the database table
Table name, leave blank to use input name Pasirinktinis	TABLE	[tekstas] Numatytas: ‚‘	Defines a name for the table that will be imported into the database. By default the table name is the name of the input vector file.
Primary Key (new field) Pasirinktinis	PK	[tekstas] Default: ‚id‘	Defines which attribute field will be the primary key of the database table
Primary Key (existing field, used if the above option is left empty) Pasirinktinis	PRIMARY_KEY	[tablefield: any] Numatytas: jokio	Defines which attribute field in the exported layer will be the primary key of the database table

continues on next page

Table 24.1000 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Geometrijos stulpelio pavadinimas Pasirinktinis	`GEOCOLUMN`	[tekstas] Numatytas: ‚geom‘	Defines in which attribute field to store the geometry information
Vector dimensions Pasirinktinis	`DIM`	[sąrašas] Default: 0 (2D)	Defines the coordinate dimensions of the imported vector data. One of: • 0 — 2 (`XY`) • 1 — 3 (`XYZ`) • 2 — 4 (`XYZM`)
Distance tolerance for simplification Pasirinktinis	`SIMPLIFY`	[tekstas] Numatytas: ‚‘	Defines a distance tolerance for the simplification of the vector geometries to be imported. By default there is no simplification.
Maximum distance between 2 nodes (densification) Pasirinktinis	`SEGMENTIZE`	[tekstas] Numatytas: ‚‘	The maximum distance between two nodes. Used to create intermediate points. By default there is no densification.
Select features by extent (defined in input layer CRS) Pasirinktinis	`SPAT`	[apimtis] Numatytas: jokio	You can select features from a given extent that will be in the output table. Galimi metodai yra: • Skaičiuoti pagal sluoksnį…: naudoja dabartiniame projekte įkelto sluoksnio apimtį • Skaičiuoti pagal išdėstymo žemėlapį…: naudoja aktyvaus projekto *išdėstymo žemėlapio elemento* apimtį • Skaičiuoti pagal žymelę…: naudoja įrašytos *žymelės* apimtį • Naudoti žemėlapio drobės apimtį • Braižyti drobėje: spauskite ir tempkite stačiakampį, apibrėžianti norimą plotą • Įveskite koordinates kaip `xmin, xmax, ymin, ymax`
Clip the input layer using the above (rectangle) extent	`CLIP`	[loginis] Numatytas: ne	The input layer will be clipped by the extent you defined before
Fields to include (leave empty to use all fields) Pasirinktinis	`FIELDS`	[string] [list] Default: []	Defines fields to keep from the imported vector file. If none is selected, all the fields are imported.
Select features using a SQL „WHERE“ statement (Ex: column=“value“) Pasirinktinis	`WHERE`	[tekstas] Numatytas: ‚‘	Defines with a SQL „WHERE“ statement which features should be selected for the output table

continues on next page

Table 24.1000 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Group N features per transaction (Default: 2000) Pasirinktinis	GT	[tekstas] Numatytas: ‚'	You can group the input features in transactions where N defines the size. By default N limits the transaction size to 20000 features.
Overwrite existing table	OVERWRITE	[loginis] Numatytas: Taip	If there is a table with the same name in the database, and if this option is set to True, the table will be overwritten.
Append to existing table	APPEND	[loginis] Numatytas: ne	If checked / True the vector data will be appended to an existing table. New fields found in the input layer are ignored. By default a new table will be created.
Append and add new fields to existing table	ADDFIELDS	[loginis] Numatytas: ne	If activated the vector data will be appended to an existing table, there won't be created a new table. New fields found in input layer are added to the table. By default a new table will be created.
Do not launder columns/table names	LAUNDER	[loginis] Numatytas: ne	With this option checked you can prevent the default behaviour (converting column names to lowercase, removing spaces and other invalid characters).
Do not create Spatial Index	INDEX	[loginis] Numatytas: ne	Prevents a spatial index for the output table from being created. By default, a spatial index is added.
Continue after a failure, skipping the failed feature	SKIPFAILURES	[loginis] Numatytas: ne	Continue after a failure, skipping the failed feature.
Validate geometries based on Simple Features specification	MAKEVALID	[loginis] Numatytas: ne	Applies cleanup operations on geometries to ensure they are valid regarding the rules of the Simple Features specification.
Promote to Multipart	PROMOTETOMULTI	[loginis] Numatytas: Taip	Casts features geometry type to multipart in the output table
Keep width and precision of input attributes	PRECISION	[loginis] Numatytas: Taip	Avoids modifying column attributes to comply with input data
Papildomos kūrimo parinktys Pasirinktinis	CREATION_OPTIO (for QGIS <= 3.42, this was OPTIONS)	[tekstas] Numatytas: ‚' (jokių papildomų parinkčių)	Papildomos GDAL kūrimo parinktys.

Išvestys

This algorithm has no output.

Pythono kodas

Algorithm ID: `gdal:importvectorintopostgisdatabasenewconnection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Vector Information

Creates an information file that lists information about an OGR-supported data source. The output will be shown in a ‚Result' window and can be written into a HTML-file. The information includes the geometry type, feature count, the spatial extent, the projection information, the list of attributes and related type, list of relations and field domains, and many more.

This algorithm is derived from the GDAL ogrinfo utility.

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis
Enable listing of all layers in the dataset	`ALL_LAYERS`	[loginis] Numatytas: ne	If checked, QGIS will output the information of all the layers in the input dataset instead of information of the first layer only. Convenient when a multi layer format (GeoPackage, GML, KML, SpatiaLite, SQLite, …) is used as input.
Summary output only	`SUMMARY_ONLY`	[loginis] Numatytas: Taip	Suppress listing of individual features and show only summary information like projection, schema, feature count and extents.
Suppress metadata info	`NO_METADATA`	[loginis] Numatytas: ne	Suppress metadata printing. Some datasets may contain a lot of metadata strings.
Sluoksnio informacija	`OUTPUT`	[html] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output HTML file that includes the file information. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą… If no HTML-file is defined the output will be written to a temporary file

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Additional command line parameters Pasirinktinis	`EXTRA`	[tekstas] Numatytas: ‚' (jokių papildomų parinkčių)	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sluoksnio informacija	`OUTPUT`	[html]	The output HTML-file that includes the file information.

Pythono kodas

Algorithm ID: `gdal:ogrinfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

Vector Information (JSON)

Creates an information file that lists information about an OGR-supported data source. The output will be shown in a ‚Result' window and can be written into a `.JSON` file. The information includes the geometry type, feature count, the spatial extent, the projection information, the list of attributes and related type, list of relations and field domains, and many more.

This algorithm is derived from the GDAL ogrinfo utility.

> **Dėmesio**
>
> Running this algorithm requires QGIS installed with GDAL >= 3.7 (see *Help* ► *About* menu).

Parametrai

Baziniai parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Įvesties sluoksnis	`INPUT`	[vektorius: geometrija]	Įvesties vektorinis sluoksnis

continues on next page

Table 24.1004 – tęsinys iš praeito puslapio

Užrašas	Pavadinimas	Tipas	Aprašymas
Enable listing of all layers in the dataset	`ALL_LAYERS`	[loginis] Numatytas: ne	If checked, QGIS will output the information of all the layers in the input dataset instead of information of the first layer only. Convenient when a multi layer format (GeoPackage, GML, KML, SpatiaLite, SQLite, …) is used as input.
Enable listing of features	`FEATURES`	[loginis] Numatytas: ne	
Suppress metadata info	`NO_METADATA`	[loginis] Numatytas: ne	Suppress metadata printing. Some datasets may contain a lot of metadata strings.
Sluoksnio informacija	`OUTPUT`	[file] Numatytas: `[Įrašyti į laikiną failą]`	Specify the output JSON file that includes the file information. *One of*: • Įrašyti į laikiną failą • Įrašyti į failą… If no JSON file is defined the output will be written to a temporary file

Išmanesni parametrai

Užrašas	Pavadinimas	Tipas	Aprašymas
Additional command line parameters Pasirinktinis	`EXTRA`	[tekstas] Numatytas: ‚‘ (jokių papildomų parinkčių)	Pridėkite papildomas GDAL komandinės eilutės parinktis

Išvestys

Užrašas	Pavadinimas	Tipas	Aprašymas
Sluoksnio informacija	`OUTPUT`	[file]	The output *.JSON* file that includes the file information.

Pythono kodas

Algorithm ID: `gdal:ogrinfojson`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

algoritmo id rodomas, kai užvedate pelę virš algoritmo apdorojimo įrankinėje. *parametrų žodynas* teikia parametrų pavadinimus ir reikšmes. Daugiau informacijos apie tai, kaip vykdyti apdorojimo algoritmus Pythono konsolėje rasite skyriuje *Using processing algorithms from the console*.

SKYRIUS 25

Priedai

25.1 QGIS priedai

QGIS buvo sukurtas su priedų architektūra. Tai leidžia lengvai pridėti į aplikaciją daug naujų savybių ir funkcijų. Kai kurios QGIS savybės realiai yra įgyvendintos kaip priedai.

25.1.1 Baziniai išoriniai priedai

QGIS priedai įgyvendinami kaip **Baziniai priedai** arba kaip**Išoriniai priedai**.

Bazinius priedus palaiko QGIS vystymo komanda ir jie automatiškai yra kiekvienos QGIS laidos dalis. Jei parašyti viena iš dviejų kalbų: **C++** ar **Python**.

Most of External Plugins are currently written in Python. They are stored either in the ‚Official' QGIS Repository at https://plugins.qgis.org/plugins/ or in external repositories and are maintained by the individual authors. Detailed documentation about the usage, minimum QGIS version, home page, authors, and other important information are provided for the plugins in the Official repository. For other external repositories, documentation might be available with the external plugins themselves. External plugins documentation is not included in this manual.

To install or activate a plugin, go to *Plugins* menu and select *Manage and install plugins...*. Installed external python plugins are placed under the `python/plugins` folder of the active *user profile* path.

Paths to Custom C++ plugins libraries can also be added under *Settings* ► *Options* ► *System*.

25.1.2 The Plugins Dialog

The Settings tab

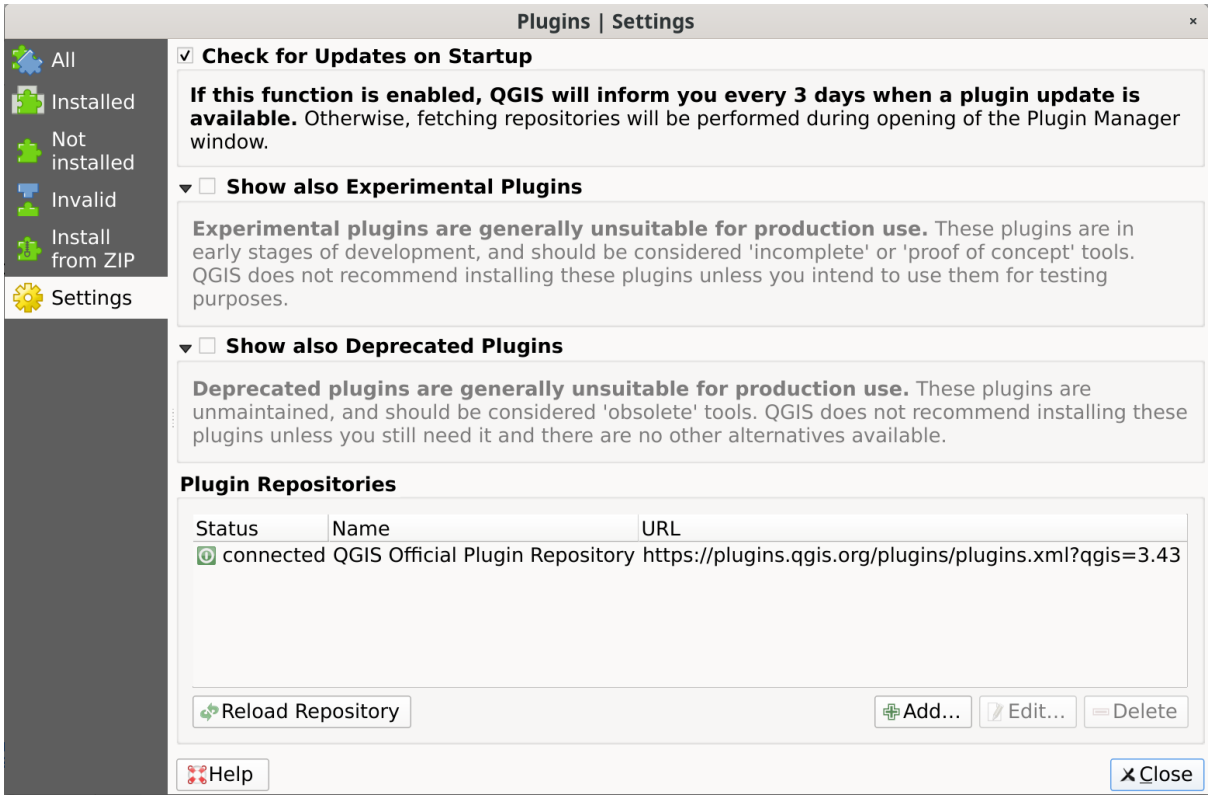
At the bottom of the left panel, the *Settings* tab is the main place you can configure which plugins can be displayed in your application. You can use the following options:

- *Check for Updates on Startup*. QGIS will automatically check for plugin updates at startup, at most once every 3 days. Whenever an installed plugin has an update available, QGIS will inform you.
- *Show also Experimental Plugins*. QGIS will show you plugins in early stages of development, which are generally unsuitable for production use. For these plugins, you can install either the stable or the experimental version, and at any moment switch from one to the other.
- *Show also Deprecated Plugins*. These plugins are usually unmaintained because they have replacement functions in QGIS, a lack of maintainers, they rely on functions that are no longer available in QGIS… They are generally unsuitable for production use and appear grayed in the plugins list.

By default, in the *Plugin Repositories* section, QGIS provides you with its official plugin repository with the URL `https://plugins.qgis.org/plugins/plugins.xml?qgis=version` (where `<version>` represents the exact QGIS version you are running). To add external author repositories, click *Add...* and fill in the *Repository Details* form with a name and the URL. The URL can be of `http://` or `file://` protocol type.

The default QGIS repository is an open repository and you don't need any authentication to access it. You can however deploy your own plugin repository and require an authentication (basic authentication, PKI). You can get more information on QGIS authentication support in *Autentifikacija* chapter.

If you do not want one or more of the added repositories, they can be disabled from the Settings tab via the *Edit...* button, or completely removed with the *Delete* button.



25.1 Fig. : The *Settings* tab

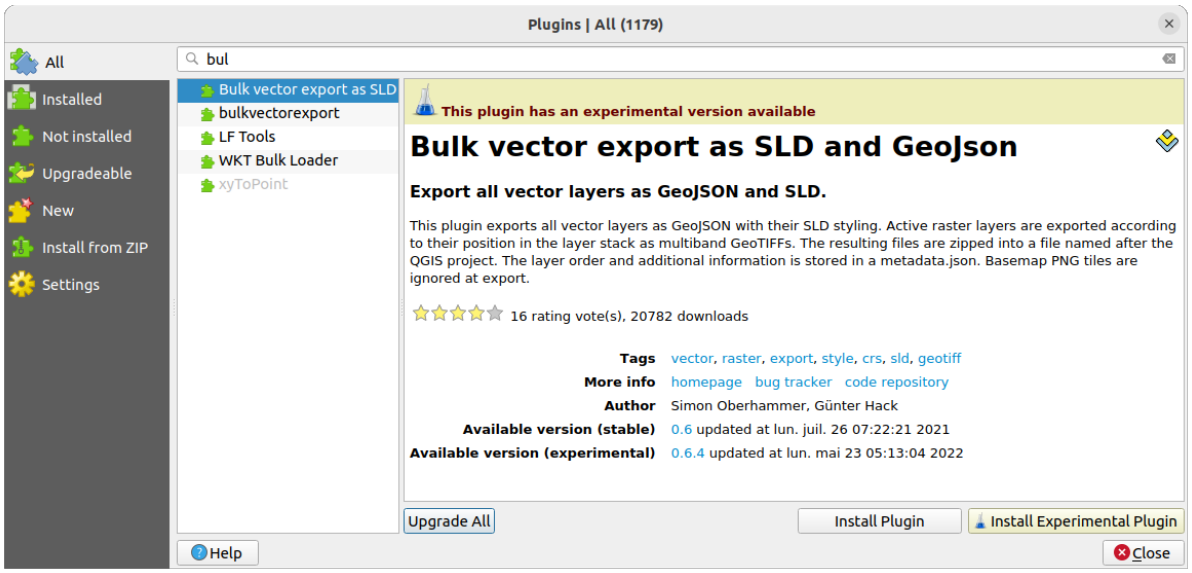
Browsing the plugins

The tabs

The upper tabs in the *Plugins* dialog provide you with lists of plugins based on their install, creation or update status. Depending on the plugins settings, available tabs can be:

- *All*: shows all the available plugins in the enabled repositories
- *Installed*: shows both the plugins you installed and the core plugins that are installed by default and you can not uninstall
- *Not installed*: shows uninstalled or not yet installed plugins in the enabled repositories
- *New*: shows plugins released since the last *Check for Updates on Startup*
- *Upgradeable*: shows installed plugins that have published a more recent version in the repository
- *Invalid*: shows all installed plugins that are currently broken for any reason (missing dependency, errors while loading, incompatible functions with QGIS version…)

At the top of the tabs, a *Search* function helps you find any plugin using metadata information (author, name, description, tag,…).



25.2 Fig. : Searching a plugin from the *All* tab

The Plugins

Select a plugin and you will have some metadata displayed in the right panel:

- information on whether the plugin is experimental or has an experimental version available (if *Show also Experimental Plugins* is checked)
- summary and description
- rating vote(s) (you can vote for your preferred plugin!)
- tags
- some useful links to the home page, tracker and code repository

- author(s)
- version(s) available with link to download page in the repository, or path to local folder for installed plugins

The *Plugin Manager* dialog allows you to interact with the latest version of the plugins. When enabled, the experimental version can be shown only if it is more recent than the latest stable version. Depending on the active tab, whether the selected plugin is installed, you will be given some of the following options:

- *Install*: installs the latest stable version of the selected plugin
- *Install Experimental Plugin*: installs the experimental version of the selected plugin
- *Reinstall Plugin*: installs the same stable version of the plugin e.g. after it has failed to load
- *Reinstall Experimental Plugin*: installs the same experimental version of the plugin e.g. after it has failed to load
- *Upgrade Plugin*: upgrades selected plugin to its latest stable version
- *Upgrade Experimental Plugin*: upgrades selected plugin to its experimental version
- *Upgrade All*: upgrades all installed plugins to their more recent stable or experimental version (depending on whether their previously installed version was stable or experimental).
- *Downgrade Plugin*: moves from the experimental version of the plugin to its previous stable version
- *Downgrade Experimental Plugin*: moves from an experimental version of the plugin to its latest published experimental version. This may occur when playing with a not yet published version.
- *Uninstall Plugin*: removes the installed plugin from the user profile

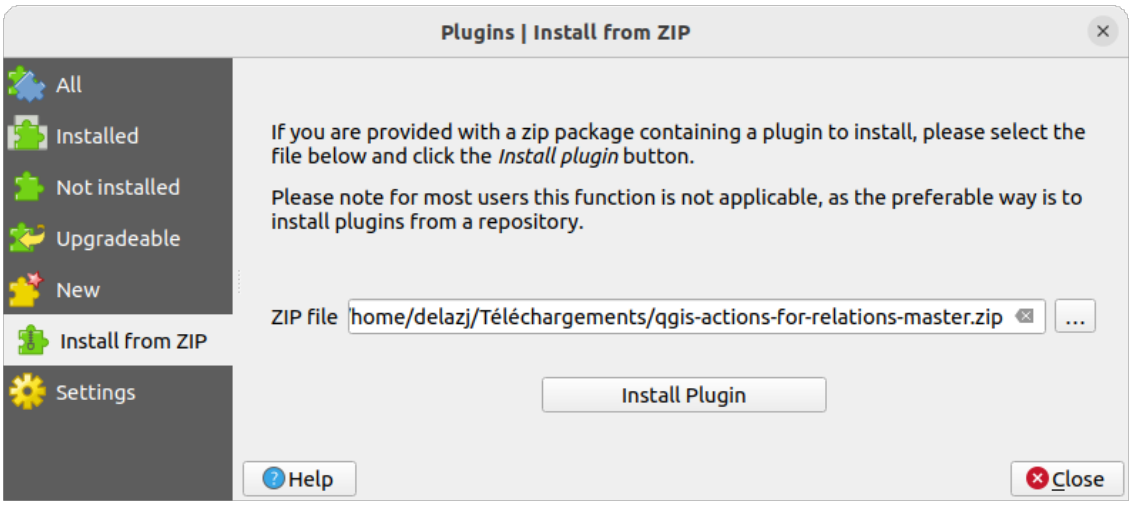
An installed plugin displays a checkbox on its left. Uncheck it to temporarily deactivate the plugin.

Right-click on a plugin in the list and you will be able to sort the plugins list by various metadata. The new order applies to all the tabs. Sort options are:

- *Sort by Name*
- *Sort by Downloads*
- *Sort by Vote*
- *Sort by Status*
- *Sort by Date Created*
- *Sort by Date Updated*

The Install from ZIP tab

The *Install from ZIP* tab provides a file selector widget to import plugins in a zipped format, e.g. plugins downloaded directly from their repository. Encrypted files are supported.

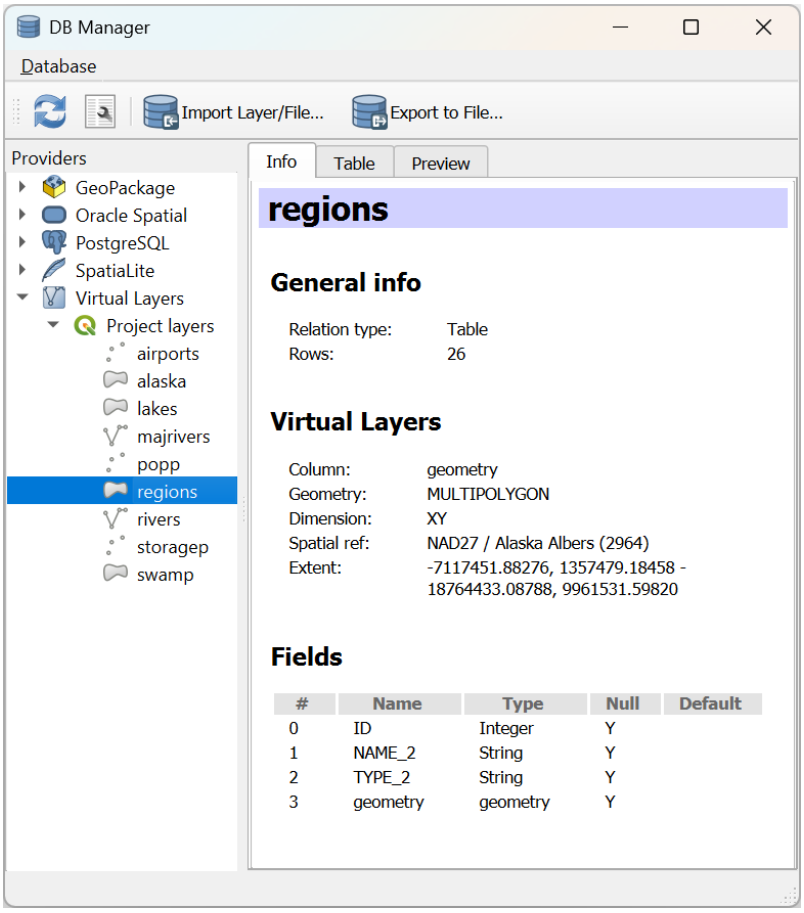


25.3 Fig. : The *Install from zip* tab

25.2 Using QGIS Core Plugins

25.2.1 DB Manager Plugin

The DB Manager Plugin is intended to be the main tool to integrate and manage spatial database formats supported by QGIS (PostgreSQL, SpatiaLite, GeoPackage, Oracle Spatial, Virtual layers) in one user interface. The DB Manager Plugin provides several features. You can drag layers from the QGIS Browser into the DB Manager, and it will import your layer into your spatial database. You can drag and drop tables between spatial databases and they will get imported.



25.4 Fig. : DB tvarkyklės dialogas

The *Database* menu allows you to connect to an existing database, to start the SQL window and to exit the DB Manager Plugin. Once you are connected to an existing database, the menus *Schema* (relevant for DBMSs, such as PostgreSQL) and *Table* will appear.

The *Schema* menu includes tools to create and delete (only if empty) schemas and, if topology is available (e.g. with PostGIS topology), to start a *TopoViewer*.

The *Table* menu allows you to create and edit tables and to delete tables and views. It is also possible to empty tables and to move tables between schemas. You can *Run Vacuum Analyze* for the selected table. *Vacuum* reclaims space and makes it available for reuse, and *analyze* updates statistics that is used to determine the most efficient way to execute a query. *Change Logging…* allows you to add change logging support to a table. Finally, you can *Import Layer/File…* and *Export to File…*.

Pastaba

Using the DB Manager it is possible to add comments for tables and columns of a PostgreSQL Database.

The *Providers* window lists all existing databases supported by QGIS. With a double-click, you can connect to the database. With the right mouse button, you can rename and delete existing schemas and tables. Tables can also be added to the QGIS canvas with the context menu.

If connected to a database, the **main** window of the DB Manager offers four tabs. The *Info* tab provides information about the table and its geometry, as well as about existing fields, constraints and indexes. It allows you to create a spatial index on a the selected table. The *Table* tab shows the table, and the *Preview* tab renders the geometries as preview. When you open an *SQL Window*, it will be placed in a new tab.

Working with the SQL Window

You can use the DB Manager to execute SQL queries against your spatial database. Queries can be saved and loaded, and there the *SQL Query Builder* will help you formulate your queries. You can even view spatial output by checking *Load as new layer* and specifying *Column(s) with unique values* (IDs), *Geometry column* and *Layer name (prefix)*. It is possible to highlight a portion of the SQL to only execute that portion when pressing `Ctrl+R` or clicking the *Execute* button.

QGIS also adds support for the `REGEXP` function in some providers. This allows users to use regular expressions in SQL filters or expressions, for example:

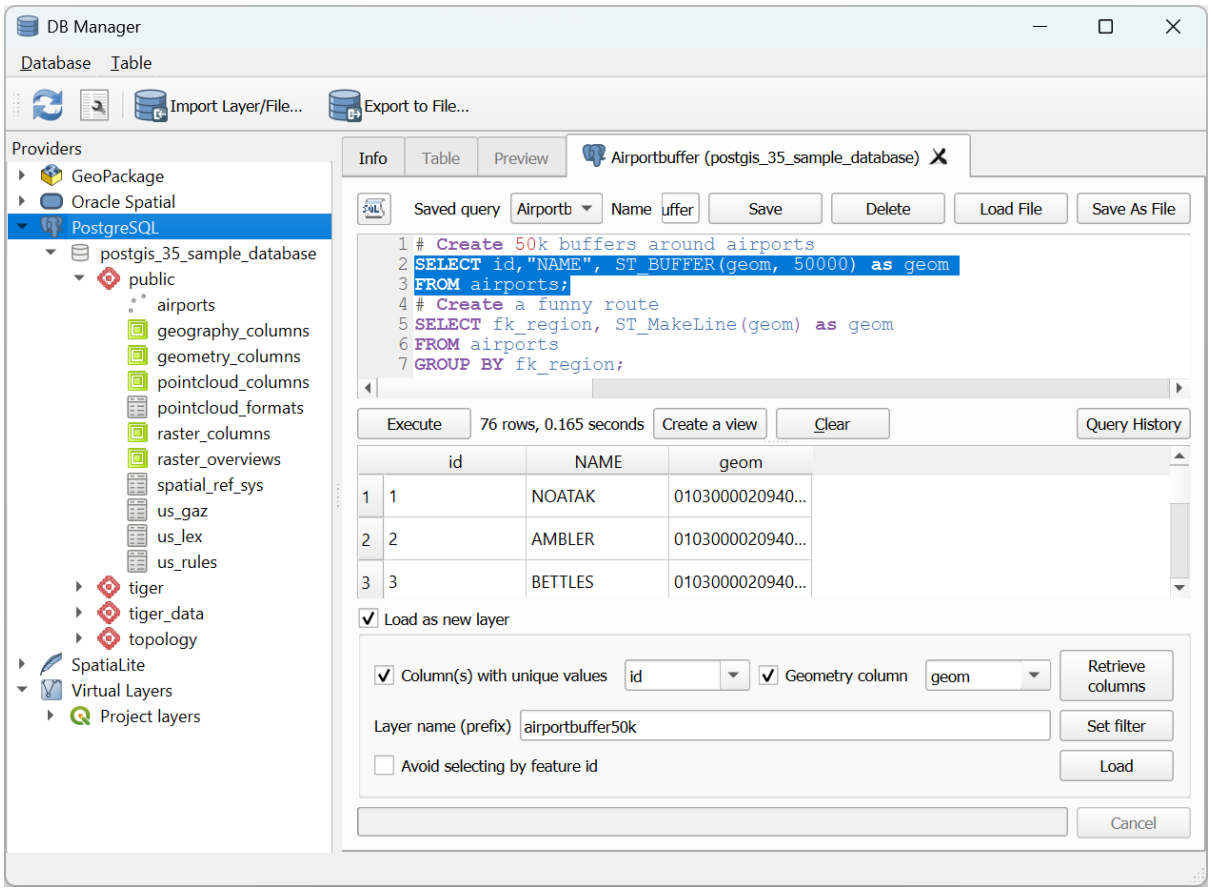
```
SELECT * FROM places WHERE name REGEXP '^A';
```

This returns all features where the `name` field starts with the letter A.

After executing your query, you can select specific cells in the result set. Use the `Ctrl+C` shortcut to copy the selected cells to the clipboard. The copied data is available as a formatted table. This allows you to paste the data into other applications, such as spreadsheet where it will show up as a table.

The *Query History* button stores the last 20 queries of each database and provider.

Double clicking on an entry will add the string to the SQL window.



25.5 Fig. : Executing SQL queries in the DB Manager SQL window

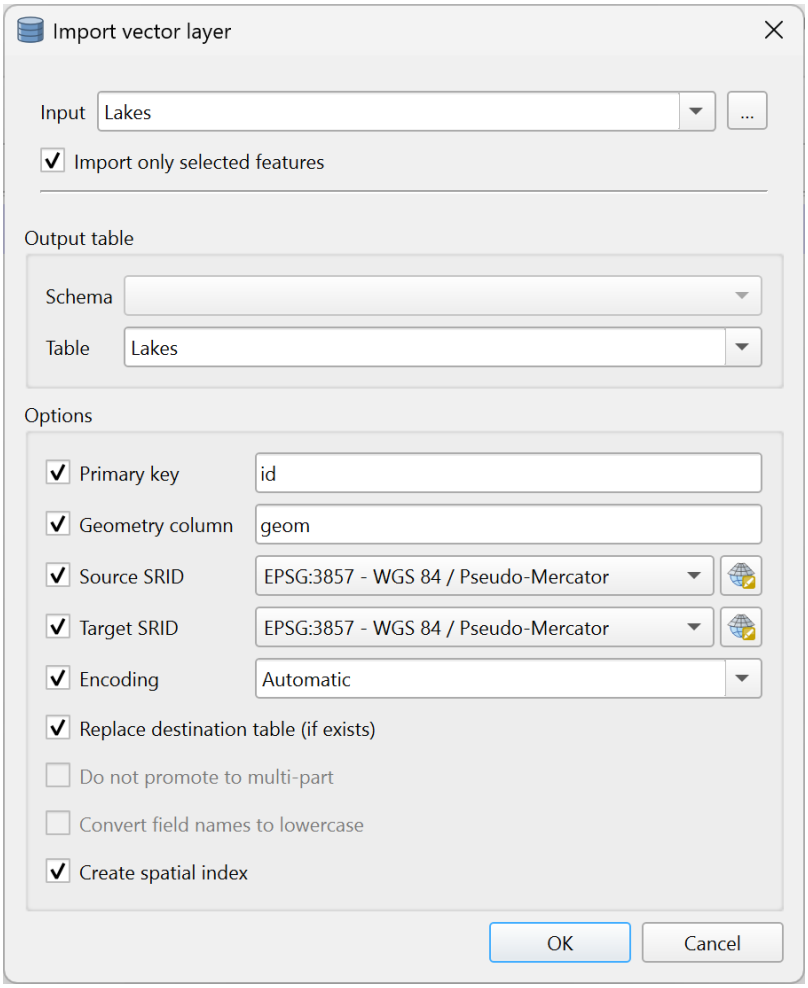
Pastaba

The SQL Window can also be used to create Virtual Layers. In that case, instead of selecting a database, select **QGIS Layers** under **Virtual Layers** before opening the SQL Window. See *Creating virtual layers* for instructions on the SQL syntax to use.

Import Vector Layer

You can import layer or file into your database. Here are the parameters you can set for the import process:

- *Input*: Select the layer or file to import. Using the dropdown menu, select from the list of loaded layers in QGIS or click on the … button to select a file from disk. Check the *Import only selected features* to import only the selected features of the layer.
- *Output table*: Choose the *Schema* and provide a name for the new table.
- *Options*: Here are some options for the import process:
  - *Primary key*: Provide naming for the primary key field. By default, it is named `id`.
  - *Geometry column*: Provide naming for the geometry column. By default, it is named `geom`.
  - *Source SRID*: Define the SRID for the geometry column. By default, it uses the layer's CRS.
  - *Target SRID*: Define the target SRID to reproject the geometries during the import process. By default, it uses the layer's CRS.
  - *Encoding*: Define the encoding of the source data. By default, it uses `Automatic`. It is QGIS's automatic detection mode that attempts to guess the file's character encoding based on available metadata or system locale.
  - *Replace destination table (if exists)*: If a table with the same name already exists in the selected schema, it will be replaced.
  - *Do not promote to multi-part*: Geometries will be imported as single-part geometries.
  - *Convert field names to lower case*: All field names will be converted to lower case.
  - *Create spatial index*: A spatial index will be created on the geometry column after import.
  - *Comment*: Add comments to table. Only available for PostgreSQL databases.

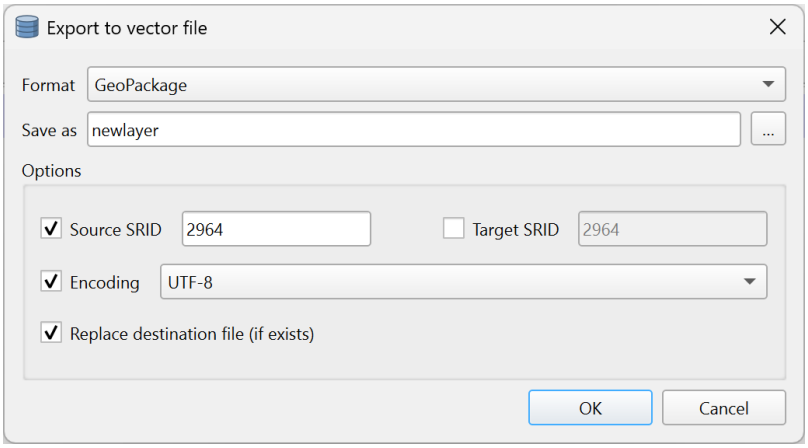


25.6 Fig. : Importing a vector layer into a spatial database using DB Manager

Export to Vector File

To export a table from your database to a vector file, select the desired *Format* and *Save as* location. Under *Options*, you can set the following parameters:

- *Source SRID*: Define the source SRID of the geometry column. By default, it uses the layer's CRS.
- *Target SRID*: Define the target SRID to reproject the geometries during the export process. By default, it uses the layer's CRS.
- *Encoding*: Define the encoding of the output data.
- *Replace destination table (if exists)*: If a table with the same name already exists at the selected location, it will be replaced.



25.7 Fig. : Exporting a table to a vector file using DB Manager

25.2.2 Geometry Checker Plugin

Geometry Checker is a powerful core plugin to check and fix the geometry validity of a layer. It is available from the *Vector* menu (*Check Geometries…*).

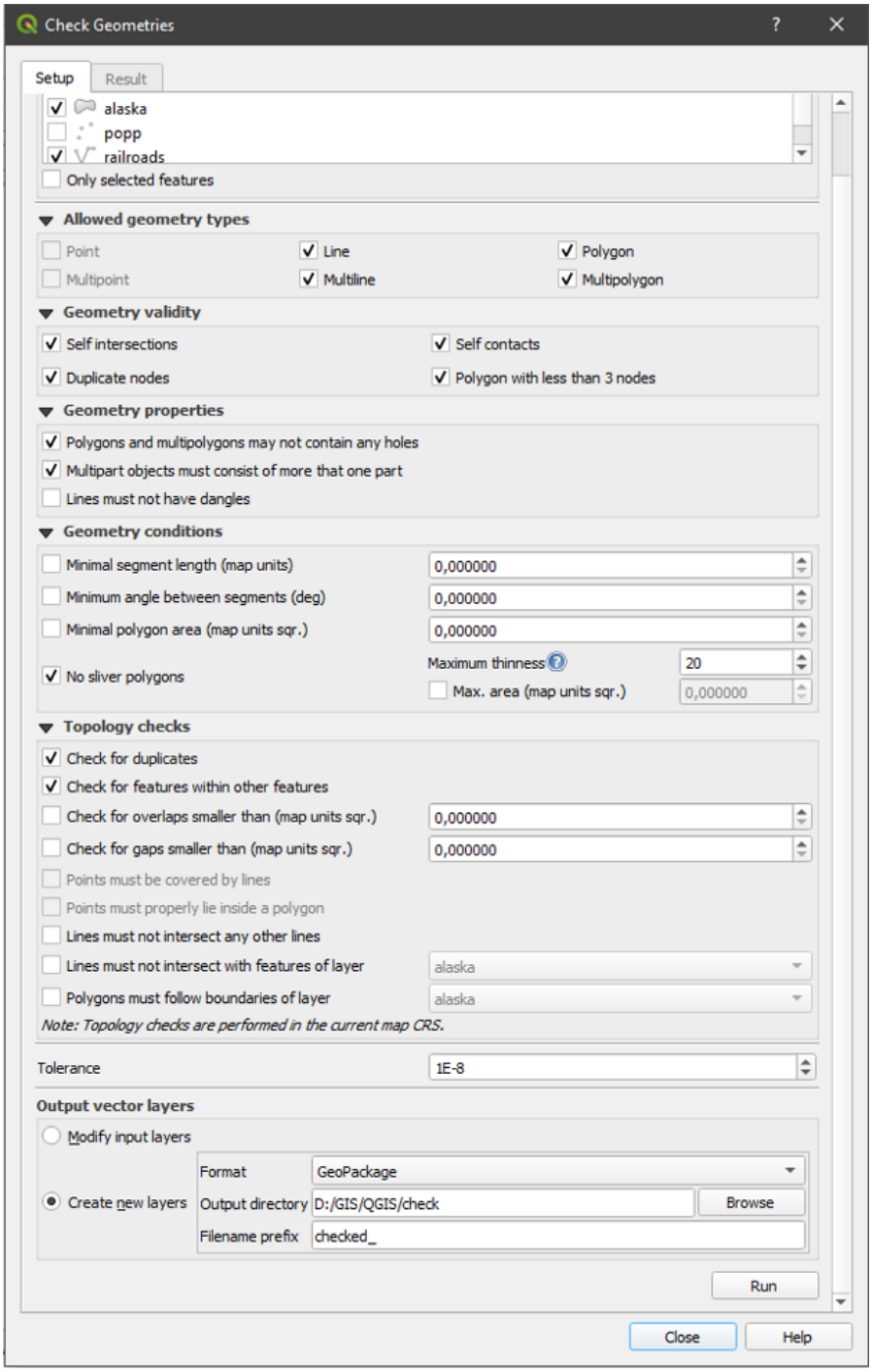
Configuring the checks

The *Check Geometries* dialog shows different grouped settings in the first tab (*Setup*):

- *Input vector layers*: to select the layers to check. A *Only selected features* checkbox can be used to restrict the checking to the geometries of the selected features.
- *Allowed geometry types* gives the chance to restrict the geometry type of the input layer(s) to:
  - Point
  - Multipoint
  - Line
  - Multiline
  - Polygon
  - Multipolygon
- *Geometry validity*. Depending on geometry types you can choose between:
  - *Self intersections*
  - *Duplicate nodes*
  - *Self contacts*
  - *Polygon with less than 3 nodes*.
- *Geometry properties*. Depending on geometry types, different options are available:
  - *Polygons and multipolygons may not contain any holes*
  - *Multipart objects must consist of more than one part*

  - *Lines must not have dangles*
- *Geometry conditions*. Allows you to add some condition to validate the geometries with:
  - *Minimal segment length (map units)* 1,00
  - *Minimum angle between segment (deg)* 1,00
  - *Minimal polygon area (map units sqr.)* 1,00
  - *No sliver polygons* with a *Maximum thinness* 1,00 and a *Max. area (map units sqr.)* 1,00
- *Topology checks*. Depending on geometry types, many different options are available:
  - *Checks for duplicates*
  - *Checks for features within other features*
  - *Checks for overlaps smaller than* 1,00
  - *Checks for gaps smaller than* 1,00
  - *Points must be covered by lines*
  - *Points must properly lie inside a polygon*
  - *Lines must not intersect any other lines*
  - *Lines must not intersect with features of layer*
  - *Polygons must follow boundaries of layer*
- *Tolerance*. You can define the tolerance of the check in map layer units.
- *Output vector layer* gives the choice to:
  - *Modify input layer*
  - *Create new layers*

When you are happy with the configuration, you can click on the *Run* button.



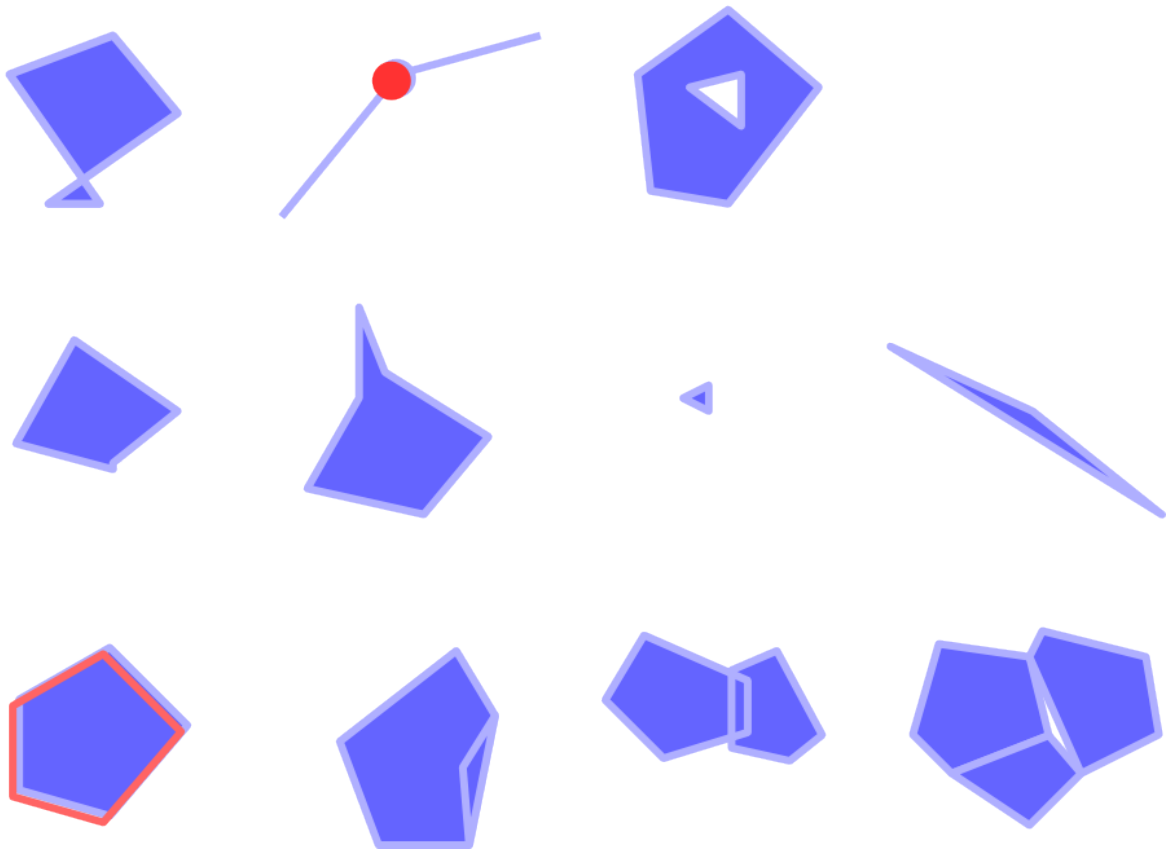
25.8 Fig. : The Geometry Checker Plugin

The *Geometry Checker Plugin* can find the following errors:

- Self intersections: a polygon with a self intersection
- Duplicate nodes: two duplicates nodes in a segment
- Holes: hole in a polygon
- Segment length: a segment length lower than a threshold
- Minimum angle: two segments with an angle lower than a threshold
- Minimum area: polygon area lower than a threshold

- Silver polygon: this error come from very small polygon (with small area) with a large perimeter
- Duplicates features
- Feature within feature
- Overlaps: polygon overlapping
- Gaps: gaps between polygons

The following figure shows the different checks made by the plugin.



25.9 Fig. : Some checks supported by the plugin

Analysing the results

The results appear in the second tab (*Result*) and as an overview layer of the errors in the canvas (its name has the default prefix `checked_`). A table lists the *Geometry check result* with one error per row and columns containing: the layer name, an ID, the error type, then the coordinates of the error, a value (depending on the type of the error) and finally the resolution column which indicates the resolution of the error. At the bottom of this table, you can *Export* the error into different file formats. You also have a counter with the number of total errors and fixed ones.

You can select a row to see the location of the error. You can change this behavior by selecting another action between *Error* (default), *Feature*, *Don't move*, and *Highlight selected features*.

Below the zoom action when clicking on the table row, you can:

- *Show selected features in attribute table*
- *Fix selected errors using default resolution*
- *Fix selected errors, prompt for resolution method* You will see a window to choose the resolution's method among which:

  - Merge with neighboring polygon with longest shared edge
  - Merge with neighboring polygon with largest area
  - Merge with neighboring polygon with identical attribute value, if any, or leave as is
  - Delete feature
  - No action
- *Error resolution settings* allows you to change the default resolution method depending on the error type

Patarimas

Fix multiple errors

You can fix multiple errors by selecting more than one row in the table with the *CTRL* + *click* action.

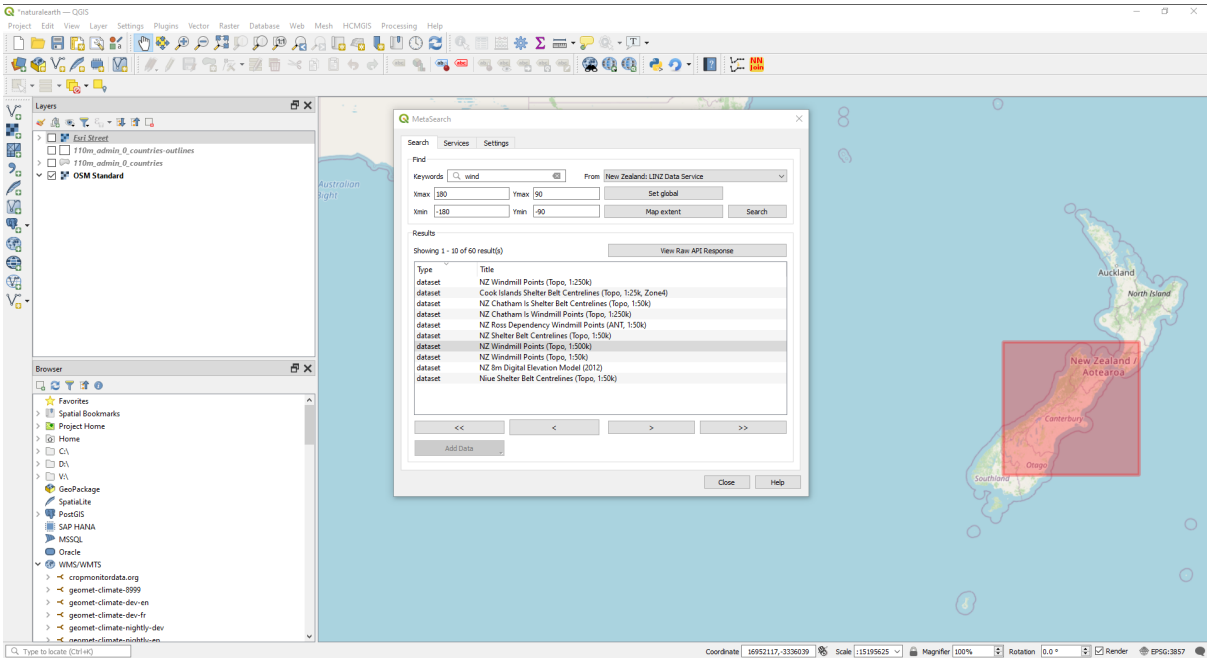
Finally, you can choose which *Attribute to use when merging features by attribute value*.

25.2.3 MetaSearch Catalog Client

Įvadas

MetaSearch is a QGIS plugin to interact with metadata catalog services, supporting both the OGC API - Records and OGC Catalog Service for the Web (CSW) standards.

MetaSearch provides an easy and intuitive approach and user-friendly interface to searching metadata catalogs within QGIS.



25.10 Fig. : Search and results of Services in MetaSearch

Working with Metadata Catalogs in QGIS

MetaSearch is included by default in QGIS, with all of its dependencies, and can be enabled from the QGIS Plugin Manager.

OGC API - Records

OGC API - Records is an OGC (Open Geospatial Consortium) standard for the discovery of geospatial resources on the Web.

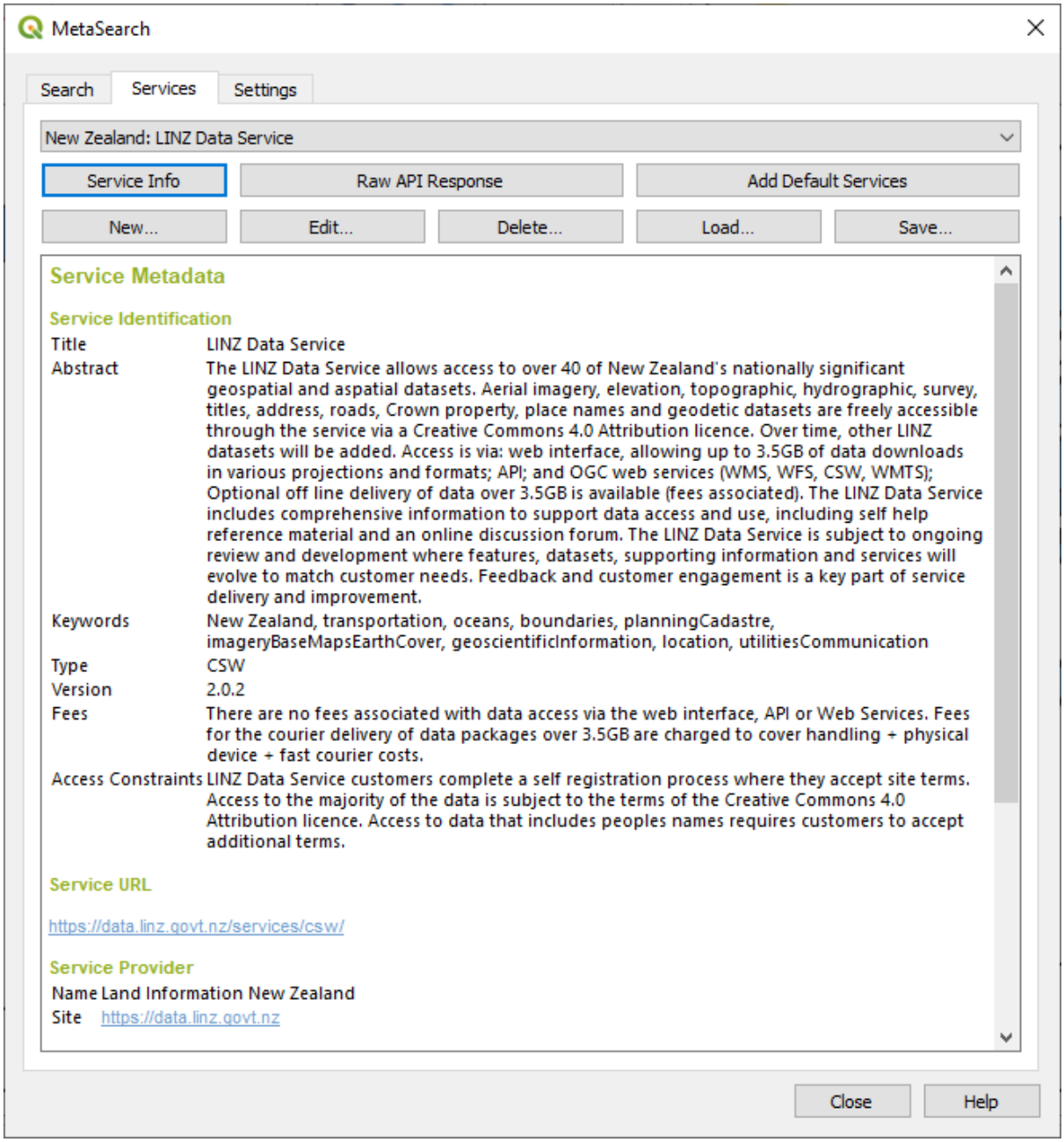
CSW (Catalog Service for the Web)

CSW (Catalog Service for the Web) is an OGC (Open Geospatial Consortium) specification that defines common interfaces to discover, browse and query metadata about data, services, and other potential resources.

Startup

To start MetaSearch, click the icon or select *Web* ► *MetaSearch* ► *MetaSearch* via the QGIS main menu. The MetaSearch dialog will appear. The main GUI consists of three tabs: *Services*, *Search* and *Settings*.

Managing Catalog Services



25.11 Fig. : Managing Catalog Services

The *Services* tab allows the user to manage all available catalog services. MetaSearch provides a default list of Catalog Services, which can be added by pressing the *Add Default Services* button.

To find all listed Catalog Service entries, click the dropdown select box.

To add a Catalog Service entry:

1. Click the *New* button
2. Enter a *Name* for the service, as well as the *URL* (endpoint). Note that for OGC CSW 2.0.2 Catalogs, only the base URL is required (not a full GetCapabilities URL). For OGC API - Records Catalogs, the URL should be the path to the collection endpoint
3. If the Catalog requires authentication, enter the appropriate *User name* and *Password* credentials.

4. Click *OK* to add the service to the list of entries.

To edit an existing Catalog Service entry:

1. Select the entry you would like to edit
2. Click the *Edit* button
3. And modify the *Name* or *URL* values
4. Spauskite *OK*.

To delete a Catalog Service entry, select the entry you would like to delete and click the *Delete* button. You will be asked to confirm deleting the entry.

MetaSearch allows loading and saving connections to an XML file. This is useful when you need to share settings between applications. Below is an example of the XML file format.

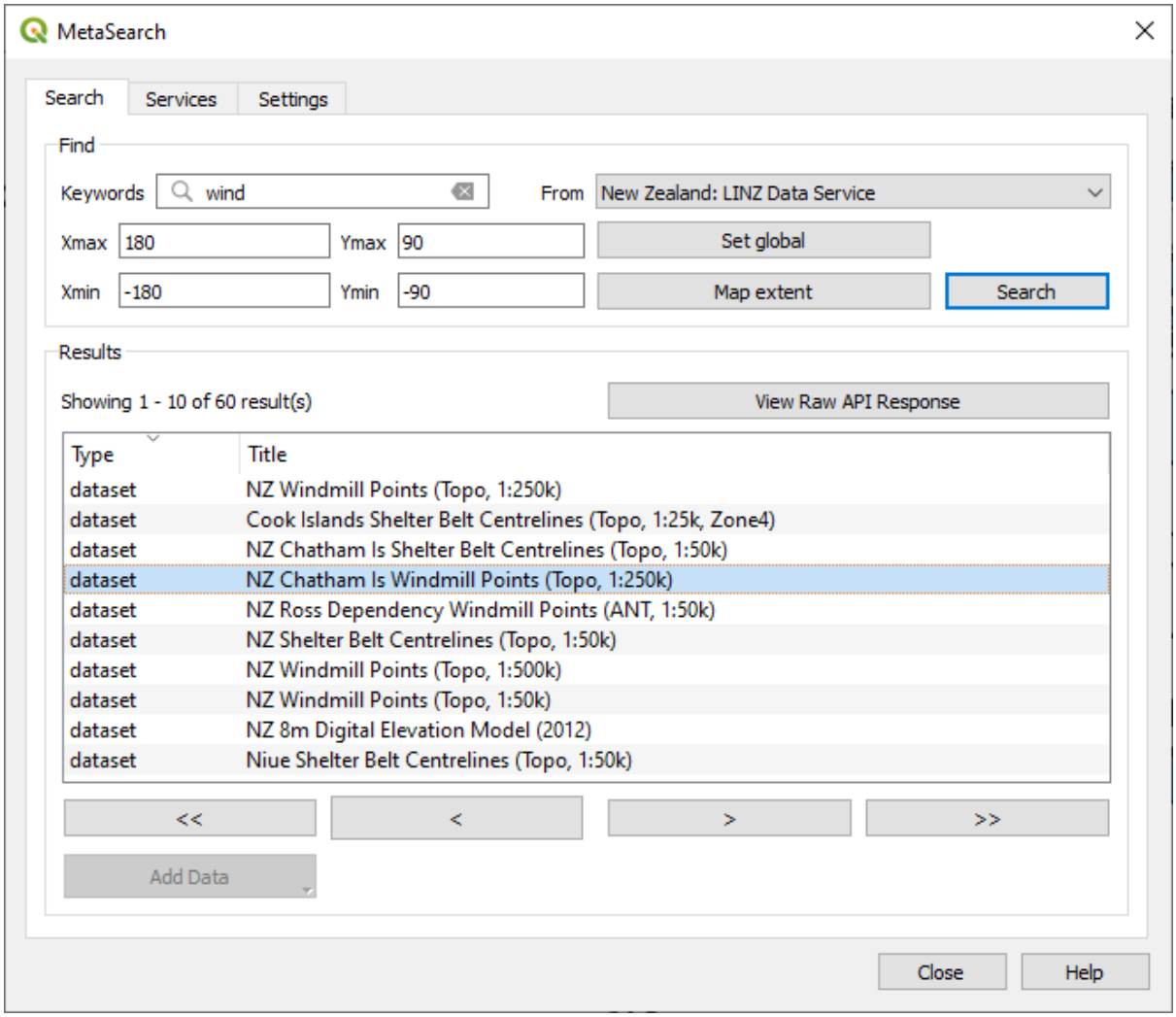
```
<?xml version="1.0" encoding="UTF-8"?>
<qgsCSWConnections version="1.0">
    <csw type="OGC CSW 2.0.2" name="Data.gov CSW" url="https://catalog.data.gov/
↪csw-all"/>
    <csw type="OGC CSW 2.0.2" name="Geonorge - National CSW service for Norway"␣
↪url="https://www.geonorge.no/geonetwork/srv/eng/csw"/>
    <csw type="OGC CSW 2.0.2" name="Geoportale Nazionale - Servizio di ricerca␣
↪Italiano" url="http://www.pcn.minambiente.it/geoportal/csw"/>
    <csw type="OGC CSW 2.0.2" name="LINZ Data Service" url="http://data.linz.govt.
↪nz/feeds/csw"/>
    <csw type="OGC CSW 2.0.2" name="Nationaal Georegister (Nederland)" url="http://
↪www.nationaalgeoregister.nl/geonetwork/srv/eng/csw"/>
    <csw type="OGC CSW 2.0.2" name="RNDT - Repertorio Nazionale dei Dati␣
↪Territoriali - Servizio di ricerca" url="http://www.rndt.gov.it/RNDT/CSW"/>
    <csw type="OGC CSW 2.0.2" name="UK Location Catalogue Publishing Service" url=
↪"http://csw.data.gov.uk/geonetwork/srv/en/csw"/>
    <csw type="OGC CSW 2.0.2" name="UNEP/GRID-Geneva Metadata Catalog" url="http://
↪metadata.grid.unep.ch:8080/geonetwork/srv/eng/csw"/>
</qgsCSWConnections>
```

To load a list of entries:

1. Click the *Load* button. A new window will appear.
2. Click the *Browse* button and navigate to the XML file of entries you wish to load.
3. Click *Open*. The list of entries will be displayed.
4. Select the entries you wish to add from the list and click *Load*.

Click the *Service Info* button to display information about the selected Catalog Service such as service identification, service provider and contact information. If you would like to view the raw API response, click the *Raw API Response* button. A separate window will open displaying server information in raw JSON or XML format.

Searching Catalog Services



25.12 Fig. : Searching catalog services

The *Search* tab allows the user to query Catalog Services for data and services, set various search parameters and view results.

The following search parameters are available:

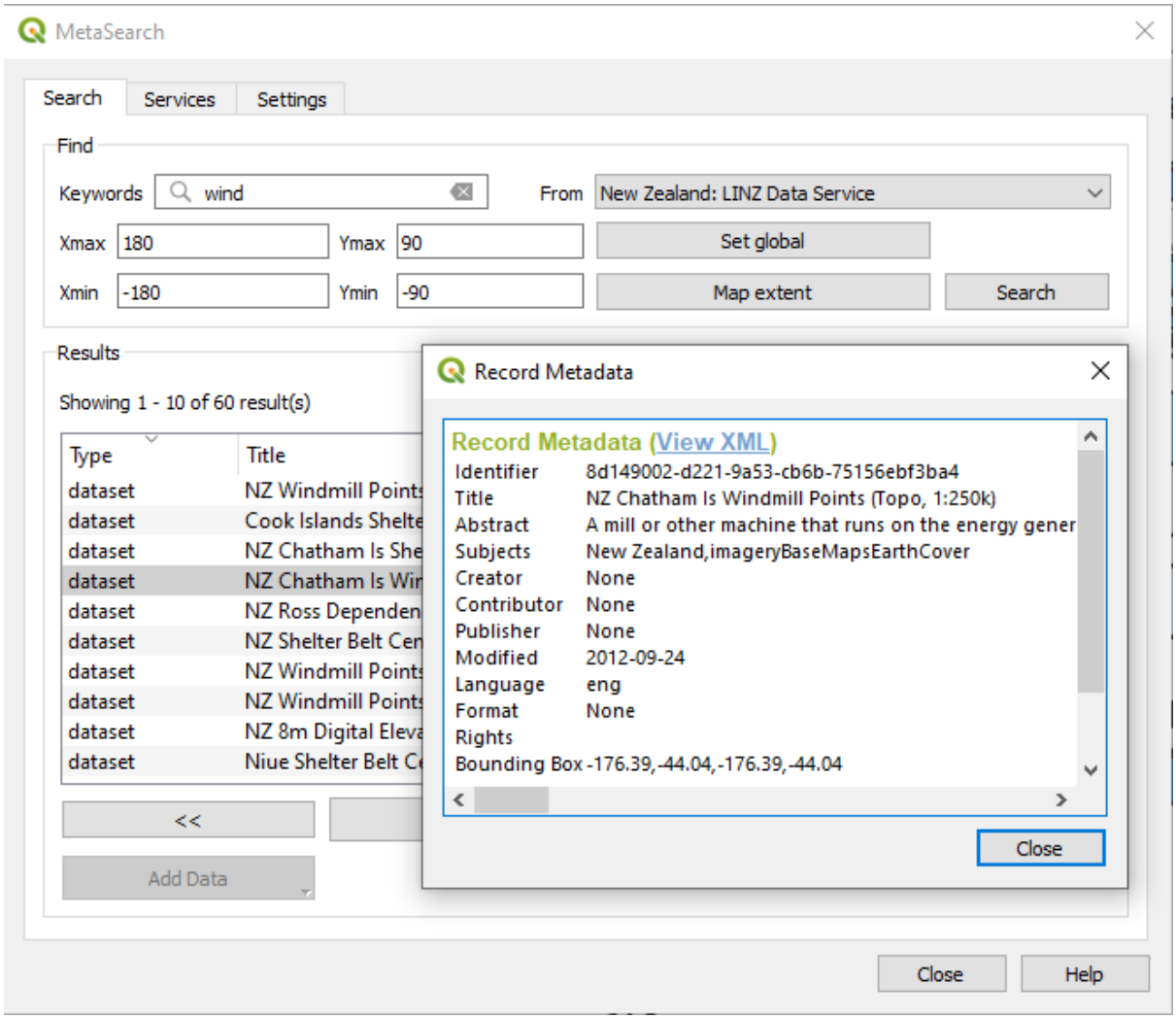
- *Keywords*: free text search keywords;
- *From*: the Catalog Service to perform the query against;
- **Bounding box**: the spatial area of interest to filter, defined by *Xmax*, *Xmin*, *Ymax*, and *Ymin*. Click *Set Global* to do a global search, click *Map Extent* to do a search in the visible area, or enter values manually.

Clicking the *Search* button will search the selected Metadata Catalog. Search results are displayed in a list, and can be sorted by clicking on the column header. You can navigate through search results with the directional buttons below the search results.

Select a result and:

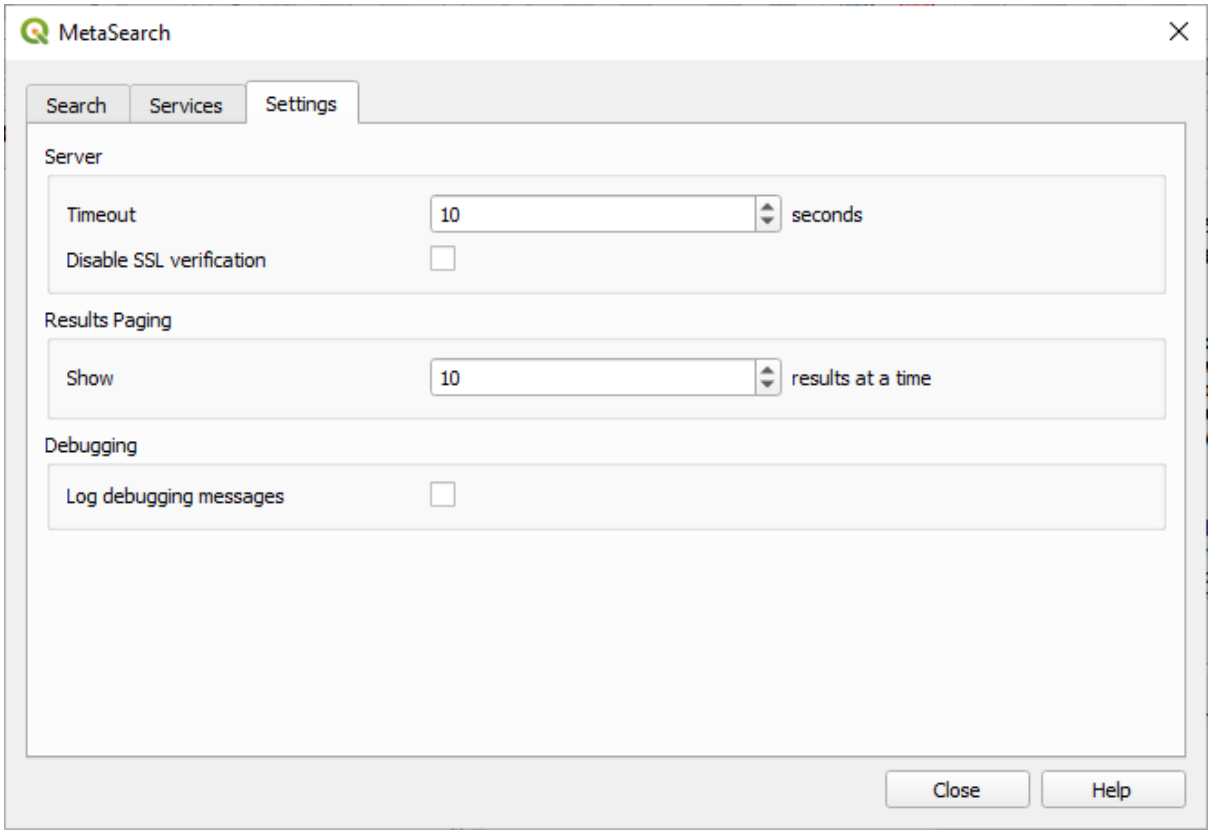
- Click the *View Raw API Response* button to open a window with the service response in raw JSON or XML format.
- If the metadata record has an associated bounding box, a footprint of the bounding box will be displayed on the map.

- Double-click the record to display the record metadata with any associated access links. Clicking a link opens the link in the user's web browser.
- If the record is a supported web service (WMS/WMTS, WFS, WCS, ArcGIS REST Service, etc.) or a GIS File, the *Add Data* button will be enabled. When clicking this button, MetaSearch will verify if this is a valid OWS. The service will then be added to the appropriate QGIS connection list, and the appropriate connection dialog will appear. If you choose *Add GIS File* the linked GIS file will be added to the current project.



25.13 Fig. : Metadata record display

Nustatymai



25.14 Fig. : MetaSearch settings

You can fine tune MetaSearch with the following *Settings*:

- *Server Timeout*: when searching metadata catalogs, the number of seconds for blocking connection attempt. Default value is 10.
- *Disable SSL verification*: option to switch off ssl verification.
- *Results paging*: when searching metadata catalogs, the number of results to show per page. Default value is 10.
- *Log debugging messages*: option to enable the logging of debugging messages in the *Pranešimų žurnalo skydelis*.

Catalog Server Errors

In some cases, the Catalog will work in a web browser, but not in MetaSearch. This may be due to the Catalog server's configuration/setup. Catalog server providers should ensure URLs are consistent and up to date in their configuration (this is common in HTTP -> HTTPS redirection scenarios). Please see the pycsw FAQ item for a deeper explanation of the issue and fix. Although the FAQ item is pycsw specific it can also apply in general to other Catalog servers.

25.2.4 Offline Editing Plugin

For data collection, it is a common situation to work with a laptop or a cell phone offline in the field. Upon returning to the network, the changes need to be synchronized with the master datasource (e.g., a PostgreSQL database). If several persons are working simultaneously on the same datasets, it is difficult to merge the edits by hand, even if people don't change the same features.

The Offline Editing Plugin automates the synchronisation by copying the content of the datasource to a SpatiaLite or GeoPackage database and storing the offline edits to dedicated tables. After being connected to the network again, it is possible to apply the offline edits to the master dataset.

To use the plugin:

1. Open a project with some vector layers (e.g., from an Esri Shapefile, PostgreSQL or WFS-T datasource).
2. Assuming you have already enabled the plugin (see *Baziniai išoriniai priedai*) go to *Database* ► *Offline Editing* ► *Convert to offline project*. The eponym dialog opens.
3. Select the *Storage type*. It can be of *GeoPackage* or *SpatiaLite* database type.
4. Use the *Browse* button to indicate the location of the database in which to store the *Offline data*. It can be an existing file or one to create.
5. In the *Select remote layers* section, check the layers you'd like to save. The content of the layers is saved to database tables.

   Pastaba

   Since target database formats do not have native list support, the offline editing plugin transforms {string, number} list fields into string fields where values are separated by commas. This allows reading and edit of the contents of those fields when offline.

   If you would like to handle both the field from the original layer and the offline layer, you can rely on the *try()* and *array* expression functions, e.g.:

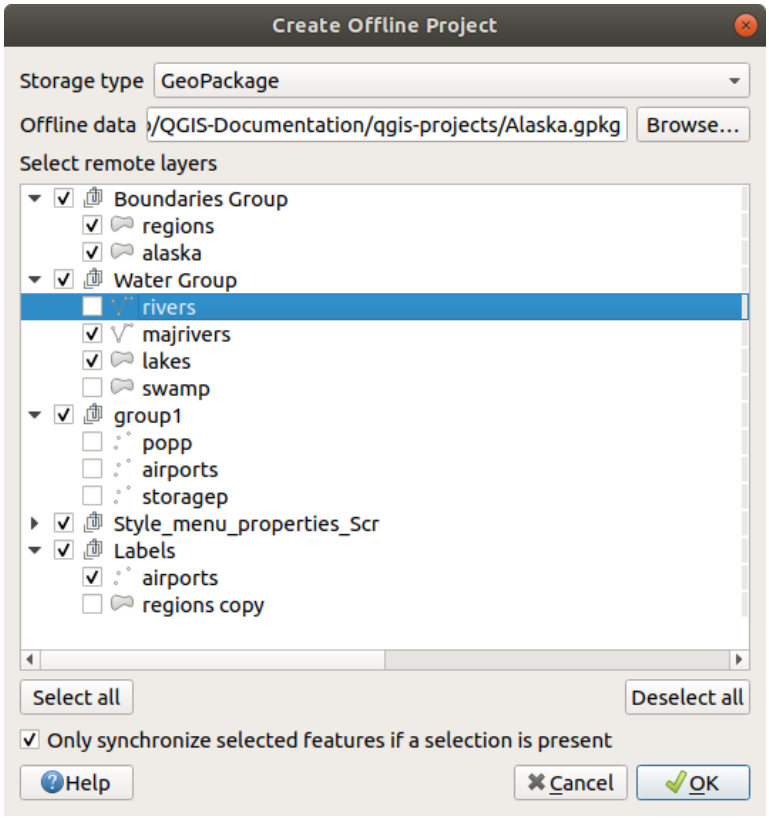
   ```
   try(array_contains("field",1),array_contains(string_to_array("field"),1))
   ```

6. You can check *Only synchronize selected features if a selection is present* allowing to only save and work on a subset. It can be invaluable in case of large layers.

   This is all!
7. Save your project and bring it on the field.
8. Edit the layers offline.
9. After being connected again, upload the changes using *Database* ► *Offline Editing* ► *Synchronize*.

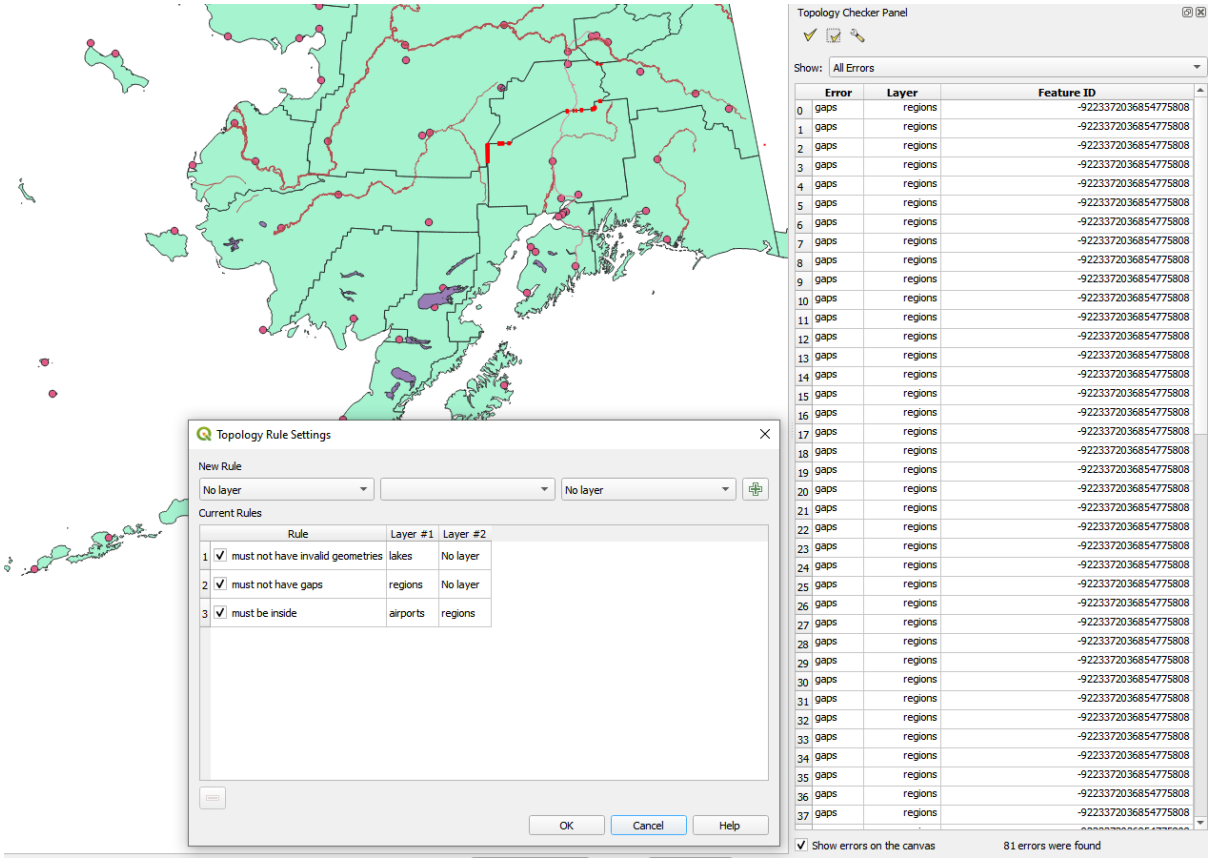
Pastaba

Layers that are used offline are marked with the icon in the *Layers* panel.



25.15 Fig. : Create an offline project

25.2.5 Topology Checker Plugin



25.16 Fig. : The Topology Checker Plugin

Topology describes the relationships between points, lines and polygons that represent the features of a geographic region. With the Topology Checker plugin, you can look over your vector files and check the topology with several topology rules. These rules check with spatial relations whether your features ‚Equal', ‚Contain', ‚Cover', are ‚CoveredBy', ‚Cross', are ‚Disjoint', ‚Intersect', ‚Overlap', ‚Touch' or are ‚Within' each other. It depends on your individual questions which topology rules you apply to your vector data (e.g., normally you won't accept overshoots in line layers, but if they depict dead-end streets you won't remove them from your vector layer).

QGIS has a built-in topological editing feature, which is great for creating new features without errors. But existing data errors and user-induced errors are hard to find. This plugin helps you find such errors through a list of rules.

To enable the *Topology checker* plugin:

1. Go to *Plugins* menu
2. Open *Manage and Install plugins* and choose *Installed*
3. Enable *Topology checker*
4. Close the *Plugin manager* dialog. A *Topology checker* entry is added to the *Vector* menu.

After you enabled *Topology checker* open it and choose *Configure* to create your topology rules.

On **point layers** the following rules are available:

- *Must be covered by*: Here you can choose a vector layer from your project. Points that aren't covered by the given vector layer occur in the ‚Error' field.
- *Must be covered by endpoints of*: Here you can choose a line layer from your project.

- *Must be inside*: Here you can choose a polygon layer from your project. The points must be inside a polygon. Otherwise, QGIS writes an ‚Error' for the point.
- *Must not have duplicates*: Whenever a point is represented twice or more, it will occur in the ‚Error' field.
- *Must not have invalid geometries*: Checks whether the geometries are valid.
- *Must not have multi-part-geometries*: All multi-part points are written into the ‚Error' field.

On **line layers**, the following rules are available:

- *End points must be covered by*: Here you can select a point layer from your project.
- *Must not have dangles*: This will show the overshoots in the line layer.
- *Must not have duplicates*: Whenever a line feature is represented twice or more, it will occur in the ‚Error' field.
- *Must not have invalid geometries*: Checks whether the geometries are valid.
- *Must not have multi-part geometries*: Sometimes, a geometry is actually a collection of simple (single-part) geometries. Such a geometry is called multi-part geometry. If it contains just one type of simple geometry, we call it multi-point, multi-linestring or multi-polygon. All multi-part lines are written into the ‚Error' field.
- *Must not have pseudos*: A line geometry's endpoint should be connected to the endpoints of two other geometries. If the endpoint is connected to only one other geometry's endpoint, the endpoint is called a pseudo node.

On **polygon layers**, the following rules are available:

- *Must contain*: Polygon layer must contain at least one point geometry from the second layer.
- *Must not have duplicates*: Polygons from the same layer must not have identical geometries. Whenever a polygon feature is represented twice or more it will occur in the ‚Error' field.
- *Must not have gaps*: Adjacent polygons should not form gaps between them. Administrative boundaries could be mentioned as an example (US state polygons do not have any gaps between them…).
- *Must not have invalid geometries*: Checks whether the geometries are valid. Some of the rules that define a valid geometry are:
  - Polygon rings must close.
  - Rings that define holes should be inside rings that define exterior boundaries.
  - Rings may not self-intersect (they may neither touch nor cross one another).
  - Rings may not touch other rings, except at a point.
- *Must not have multi-part geometries*: Sometimes, a geometry is actually a collection of simple (single-part) geometries. Such a geometry is called multi-part geometry. If it contains just one type of simple geometry, we call it multi-point, multi-linestring or multi-polygon. For example, a country consisting of multiple islands can be represented as a multi-polygon.
- *Must not overlap*: Adjacent polygons should not share common area.
- *Must not overlap with*: Adjacent polygons from one layer should not share common area with polygons from another layer.

When you create a *New rule* click on the Add rule to include it to the *Current rules*. You can enable or disable individual rules by clicking on the checkbox. Right-clicking over a rule provides the following options:

- *Select All* the rules
- *Activate* or *Deactivate* the selected rules
- *Toggle activation* of selected rules
- *Delete* selected rules. This can also be achieved with the Delete selected rules button.

Press *OK* and then choose from the *Topology checker* panel:

- Validate All: applies the active rules to all the features of the involved layer(s)
- or Validate Extent: applies the active rules to the features of the involved layer(s), within the current map canvas. The button is kept pushed and the results will update as the map canvas extent changes.

Errors will show up in the table of results containing type of error, layer and feature ID. Use *Filter errors by rule* menu to filter the errors to a specific error type.

Check *Show errors on the canvas* to show error location on the canvas. Clicking a row in the table will zoom the map canvas to the concerned feature, where you can use *QGIS digitizing tools* to fix the error.

Below is the list of Core plugins provided with QGIS. They are not necessarily enabled by default.

Piktograma	Plugin	Aprašymas	Manual Reference
	DB tvarkyklė	Manage your databases within QGIS	*DB Manager Plugin*
	Geometry Checker	Check and repair errors in vector geometries	*Geometry Checker Plugin*
	GRASS 7	GRASS functionality	*GRASS GIS Integration*
	GRASS GIS provider	GRASS GIS Processing functionality	*GRASS GIS Integration*
	MetaSearch Catalog Client	Interact with metadata catalog services (CSW)	*MetaSearch Catalog Client*
	Offline Editing	Offline editing and synchronizing with database	*Offline Editing Plugin*
	Apdorojimas	Spatial data processing framework	*QGIS processing framework*
	Topology Checker	Find topological errors in vector layers	*Topology Checker Plugin*

Pastaba

To use the Core Plugins GRASS 7, GRASS GIS provider, they have to be configured. Informations can be found *here*.

25.3 QGIS Python konsolė

Kaip pamatysite vėliau šiame skyriuje, QGIS buvo sukurtas su priedų architektūra. Priedus galima rašyti su Python, geoerdviniame pasaulyje labai gerai žinoma kalba.

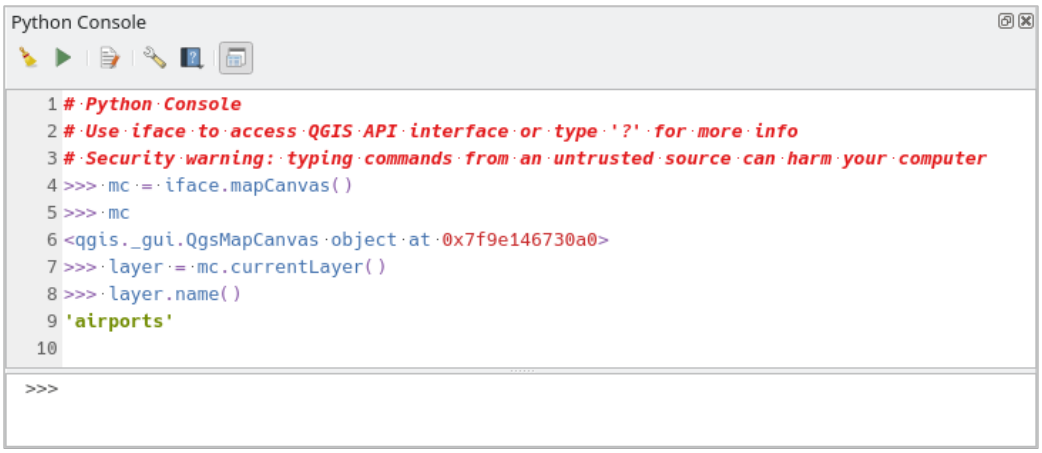
QGIS suteikia Python API (kodo pavyzdžių rasite PyQGIS programuotojo receptų knygoje), leidžiančią naudotojui sąveikauti su jo objektais (sluoksniais, geoobjektais ar naudotojo sąsaja). QGIS taipogi turi Python konsolę.

The QGIS Python Console is an interactive shell for Python command executions. It also has a Python file editor that allows you to edit and save your Python scripts. Both console and editor are based on PyQScintilla2 package. To open the console go to *Plugins* ► *Python Console* (`Ctrl+Alt+P`) or click on the Python Console icon in the *Plugins* toolbar.

25.3.1 The Interactive Console

The console is a Python interpreter that allows you to execute Python commands. Modules from QGIS (analysis, core, gui, server, processing, 3d) and Qt (QtCore, QtGui, QtNetwork, QtWidgets, QtXml) as well as Python's math, os, re and sys modules are already imported and can be used directly.

The interactive console is composed of a toolbar, an input area and an output one.



25.17 Fig. : The Python Console

Įrankinė

The toolbar proposes the following tools:

- Clear Console to wipe the output area;
- Run Command available in the input area: same as pressing `Enter`;
- Show Editor: toggles *The Code Editor* visibility;
- Options...: opens a dialog to configure *console properties*;
- Help... provides a menu to access various documentation:
  - *Python Console Help* (the current page)
  - PyQGIS API documentation
  - PyQGIS Cookbook
- Dock Code Editor to dock or undock the panel in QGIS interface

Input area

The Console input area main features are:

- Code completion, highlighting syntax and calltips for the following APIs:
  - Python
  - PyQGIS
  - PyQt5
  - QScintilla2
  - osgeo-gdal-ogr

  - osgeo-geos

- `Ctrl+Alt+Space` to view the auto-completion list if enabled in the *Pythono nustatymai*;
- Execute code snippets from the input area by typing and pressing `Enter` or *Run Command*;
- Execute code snippets from the output area using the *Enter Selected* from the contextual menu or pressing `Ctrl+E`;
- Browse the command history from the input area using the `Up` and `Down` arrow keys and execute the command you want;
- `Ctrl+Shift+Space` to view the command history: double-clicking a row will execute the command. The *Command History* dialog can also be accessed from context menu of input area;
- Save and clear the command history. The history will be saved into the `console_history.txt` file under the active *user profile* folder;
- Type the following special commands:
  - `?` to show a help of the Python Console
  - `_api` to open QGIS C++ API documentation or `_api(object)` for a specific object documentation (in QGIS C++ API or Qt API documentation)
  - `_pyqgis` to open QGIS Python API documentation or `_pyqgis(object)` for a specific object documentation (in QGIS Python API or Qt API documentation)
  - `_cookbook` to open PyQGIS Cookbook.
  - `!` followed by a command to execute Shell commands from the Python Console. The console will start a subprocess, and forward its output to the Python Console Output. While the subprocess is running, the Python Console Input switches to STDIN mode and forwards entered character to the child process. This makes it possible to send confirmation when the child program asks for it. When the Console is in STDIN mode, pressing `Ctrl+C` kills the subprocess. It is also possible to affect the result of a command to a variable with the syntax `var = !cmd`

```
>>> !echo QGIS Rocks!
QGIS Rocks

>>> !gdalinfo --version
GDAL 3.10.3, released 2025/04/01

>>> !pip install black
# Install black python formatter using pip (if available)

>>> sql_formats = !ogrinfo --formats | grep SQL
>>> sql_formats
['SQLite -vector- (rw+v): SQLite / Spatialite', '  MSSQLSpatial -vector-␣
↪(rw+): Microsoft SQL Server Spatial Database', '  PostgreSQL -vector-␣
↪(rw+): PostgreSQL/PostGIS', '  MySQL -vector- (rw+): MySQL', '  PGDUMP -
↪vector- (w+v): PostgreSQL SQL dump']
```

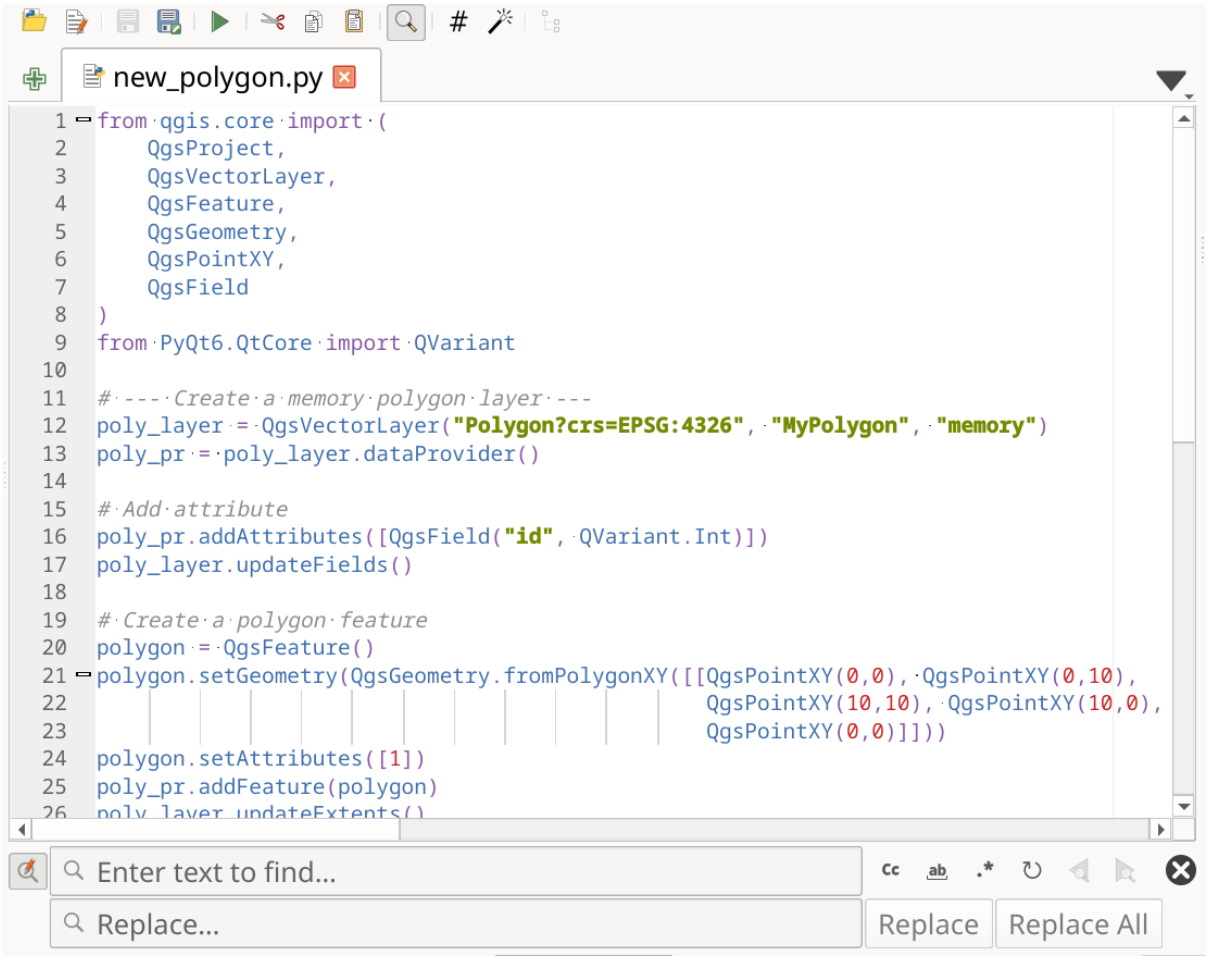
Patarimas

Reuse executed commands from the output panel

You can execute code snippets from the output panel by selecting some text and pressing `Ctrl+E`. No matter if selected text contains the interpreter prompt (`>>>`, `...`).

25.3.2 The Code Editor

Use the Show Editor button in the Interactive Console to enable the editor widget. It allows editing and saving Python scripts and offers advanced functionalities to manage your code. Depending on the *enabled settings*, it provides various capabilities for easier code writing, such as code completion, highlighting syntax and calltips for supported APIs. Automatic indentation, parenthesis insertion, code commenting and syntax checking are also available.



25.18 Fig. : The Python Console editor

The code editor area allows to simultaneously work on different scripts, each in a specific tab. Press New editor to add a new tab. You can run partially or totally a script from the *Code Editor* and output the result in the Interactive Console output area.

Patarimas

Press `Ctrl+Space` to view the auto-completion list.

At the top of the dialog, a toolbar provides access to a few commands. Right-clicking the editor area provides some more options. All available tools are described in the following table.

25.1 Table : Python Console Editor Tools

Tool name	Function	Location
Open script…	Loads a Python file in the code editor dialog, as a new tab	Įrankinė
Open in external editor…	Opens a saved Python script in the default external program set for Python file editing	
Save (`Ctrl+S`)	Saves the current script	
Save as… (`Ctrl+Shift+S`)	Saves the current script as a new file	
Run script (`Ctrl+Shift+E`)	Executes the whole script in the Interactive console (this creates a byte-compiled file with the extension `.pyc`)	Toolbar & Contextual menu
Context help (`F1`)	Attempts to display help on the selected string (class, method, object,…) in its corresponding API documentation	Contextual menu
Run selected (`Ctrl+E`)	Executes in the Interactive console the lines selected in the script	Toolbar & Contextual menu
Cut (`Ctrl+X`)	Cuts selected text to the clipboard	
Copy (`Ctrl+C`)	Copies selected text to the clipboard	
Paste (`Ctrl+V`)	Pastes a cut or copied text	
Find text	Allows to search and replace a text in the script. • Use the default Desktop Environment shortcut to find next/previous (`Ctrl+G` and `Shift+Ctrl+G`); • Automatically find first match when typing in find box; • Set initial find string to selection when opening find; • Enable the *Use Regular Expressions* option in the search bar to use regular expressions instead of plain text matching; • Replace the matching text with the text in the replace box. When using a regular expression, you can reference captured groups (e.g. via `\1`) in the replacement text. Use *Replace All* to replace all matches at once. • Pressing `Esc` closes the find bar.	
Toggle comment (`Ctrl+:`)	Comments out or uncomment selected lines, by adding or removing `#` character (potentially followed by a space) at the start of the lines. If the selection contains both comments and actual commands, adds `#` at the start of each non empty line. Pressing again the button reverts to the initial state.	
Reformat code (`Ctrl+Alt+F`)	Allows to manually apply various formatting rules (sort imports, indentation, line length,…) to the code, following *user-defined settings*. This may require installation of additional Python modules.	
Object inspector…	Shows and hides a dedicated browser with a tree structure for classes and functions available in the script. Click an item fo a quick access to its definition. The tool requires an activation from the *Python settings - Run and Debug*.	
Hide editor	Hides the Python code editor block. To make it visible again, press Show editor button from the interactive console toolbar.	Contextual menu

continues on next page

Table 25.1 – tęsinys iš praeito puslapio

Tool name	Function	Location
Check syntax (`Ctrl+4`)	Browses the code and reports syntax errors, such as missing parenthesis, colons, wrong indentation,…	
Undo (`Ctrl+Z`)	Undoes the latest action	
Redo (`Ctrl+Shift+Z`)	Reverts undone actions to a more recent	
Select all (`Ctrl+A`)	Selects the whole script	
Share on GitHub	Shares the script as a *Secret Gist* or *Public Gist* on GitHub, provided a *GitHub access token*.	
Options…	Opens the *Pythono nustatymai* dialog.	

26.1 Appendix C: QGIS File Formats

26.1.1 QGS/QGZ - The QGIS Project File Format

The **QGS** format is an XML format for storing QGIS projects. The **QGZ** format is a compressed (zip) archive containing a QGS file and a QGD file. The **QGD** file is the associated sqlite database of the qgis project that contain auxiliary data for the project. If there are no auxiliary data, the QGD file will be empty.

A QGIS file contains everything that is needed for storing a QGIS project, including:

- project title
- project CRS
- the layer tree
- snapping settings
- relations
- the map canvas extent
- project models
- legend
- mapview docks (2D and 3D)
- the layers with links to the underlying datasets (data sources) and other layer properties including extent, SRS, joins, styles, renderer, blend mode, opacity and more.
- project properties

The figures below show the top level tags in a QGS file and the expanded `ProjectLayers` tag.

```
−<qgis version="3.4.13-Madeira" projectname="">
    <homePath path=""/>
    <title/>
    <autotransaction active="0"/>
    <evaluateDefaultValues active="0"/>
    <trust active="0"/>
  +<projectCrs></projectCrs>
  +<layer-tree-group></layer-tree-group>
  +<snapping-settings tolerance="12" unit="1" enabled="0" type="1" mode="2" intersection-snapping="0">
    </snapping-settings>
    <relations/>
  −<mapcanvas name="theMapCanvas" annotationsVisible="1">
      <units>meters</units>
    +<extent></extent>
      <rotation>0</rotation>
    +<destinationsrs></destinationsrs>
      <rendermaptile>0</rendermaptile>
    </mapcanvas>
    <projectModels/>
  +<legend updateDrawingOrder="true"></legend>
    <mapViewDocks/>
    <mapViewDocks3D/>
  +<projectlayers></projectlayers>
  +<layerorder></layerorder>
  +<properties></properties>
    <visibility-presets/>
    <transformContext/>
  +<projectMetadata></projectMetadata>
    <Annotations/>
    <Layouts/>
  </qgis>
```

26.1 Fig. : The top level tags in a QGS file

```
−<projectlayers>
 −<maplayer styleCategories="AllStyleCategories" readOnly="0" autoRefreshTime="0" autoRefreshEnabled="0" refreshOnNotifyEnabled="0" maxScale="0"
  geometry="Polygon" labelsEnabled="0" type="vector" simplifyDrawingHints="1" hasScaleBasedVisibilityFlag="0" simplifyDrawingTol="1"
  simplifyMaxScale="1" minScale="1e+8" simplifyAlgorithm="0" simplifyLocal="1" refreshOnNotifyMessage="">
  +<extent></extent>
   <id>watersheds_b62efa19_8809_4406_b6ec_2951ac4c94c5</id>
  −<datasource>
     ./QGIS-Training-Data-2.0/exercise_data/processing/generalize/watersheds.shp
   </datasource>
  +<keywordList></keywordList>
   <layername>watersheds</layername>
  +<srs></srs>
  +<resourceMetadata></resourceMetadata>
   <provider encoding="UTF-8">ogr</provider>
   <vectorjoins/>
   <layerDependencies/>
   <dataDependencies/>
   <legend type="default-vector"/>
   <expressionfields/>
  +<map-layer-style-manager current="default"></map-layer-style-manager>
   <auxiliaryLayer/>
  +<flags></flags>
  +<renderer-v2 symbollevels="0" enableorderby="0" type="singleSymbol" forceraster="0"></renderer-v2>
  +<customproperties></customproperties>
   <blendMode>0</blendMode>
   <featureBlendMode>0</featureBlendMode>
   <layerOpacity>1</layerOpacity>
  +<SingleCategoryDiagramRenderer diagramType="Histogram" attributeLegend="1"></SingleCategoryDiagramRenderer>
  +<DiagramLayerSettings priority="0" linePlacementFlags="18" dist="0" showAll="1" placement="1" obstacle="0" zIndex="0"></DiagramLayerSettings>
  +<geometryOptions removeDuplicateNodes="0" geometryPrecision="0"></geometryOptions>
  +<fieldConfiguration></fieldConfiguration>
  +<aliases></aliases>
   <excludeAttributesWMS/>
   <excludeAttributesWFS/>
  +<defaults></defaults>
  +<constraints></constraints>
  +<constraintExpressions></constraintExpressions>
   <expressionfields/>
  +<attributeactions></attributeactions>
  +<attributetableconfig actionWidgetStyle="dropDown" sortExpression="" sortOrder="0"></attributetableconfig>
  +<conditionalstyles></conditionalstyles>
   <editform tolerant="1"/>
   <editforminit/>
   <editforminitcodesource>0</editforminitcodesource>
   <editforminitfilepath/>
  +<editforminitcode></editforminitcode>
   <featformsuppress>0</featformsuppress>
   <editorlayout>generatedlayout</editorlayout>
  +<editable></editable>
  +<labelOnTop></labelOnTop>
   <widgets/>
   <previewExpression>ID</previewExpression>
   <mapTip/>
 </maplayer>
</projectlayers>
```

26.2 Fig. : The expanded top level ProjectLayers tag of a QGS file

26.1.2 QLR - The QGIS Layer Definition file

A Layer Definition file (QLR) is an XML file that contains a pointer to the layer data source in addition to QGIS style information for the layer.

The use case for this file is simple: To have a single file for opening a data source and bringing in all the related style information. QLR files also allow you to mask the underlying datasource in an easy to open file.

An example of QLR usage is for opening MS SQL layers. Rather than having to go to the MS SQL connection dialog, connect, select, load and finally style, you can simply add a .qlr file that points to the correct MS SQL layer with all the necessary style included.

In the future a .qlr file may hold a reference to more than one layer.

```
−<qlr>
  +<layer-tree-group name="" checked="Qt::Checked" expanded="1"></layer-tree-group>
  −<maplayers>
    −<maplayer autoRefreshEnabled="0" labelsEnabled="0" autoRefreshTime="0" readOnly="0" refreshOnNotifyMessage=""
     geometry="Line" simplifyDrawingTol="1" simplifyMaxScale="1" styleCategories="AllStyleCategories" simplifyDrawingHints="1"
     maxScale="0" simplifyLocal="1" hasScaleBasedVisibilityFlag="0" type="vector" refreshOnNotifyEnabled="0" minScale="1e+8"
     simplifyAlgorithm="0">
      +<extent></extent>
        <id>inputnew_6740bb2e_0441_4af5_8dcf_305c5c4d8ca7</id>
      +<datasource></datasource>
      +<keywordList></keywordList>
        <layername>inputnew</layername>
      +<srs></srs>
      +<resourceMetadata></resourceMetadata>
        <provider encoding="UTF-8">ogr</provider>
        <vectorjoins/>
        <layerDependencies/>
        <dataDependencies/>
        <legend type="default-vector"/>
        <expressionfields/>
      +<map-layer-style-manager current="default"></map-layer-style-manager>
        <auxiliaryLayer/>
      +<flags></flags>
      +<renderer-v2 enableorderby="0" type="singleSymbol" forceraster="0" symbollevels="0"></renderer-v2>
      +<customproperties></customproperties>
        <blendMode>0</blendMode>
        <featureBlendMode>0</featureBlendMode>
        <layerOpacity>1</layerOpacity>
      +<geometryOptions removeDuplicateNodes="0" geometryPrecision="0"></geometryOptions>
      +<fieldConfiguration></fieldConfiguration>
      +<aliases></aliases>
        <excludeAttributesWMS/>
        <excludeAttributesWFS/>
      +<defaults></defaults>
      +<constraints></constraints>
      +<constraintExpressions></constraintExpressions>
        <expressionfields/>
      +<attributeactions></attributeactions>
      +<attributetableconfig sortExpression="" actionWidgetStyle="dropDown" sortOrder="0"></attributetableconfig>
      +<conditionalstyles></conditionalstyles>
        <editform tolerant="1">../src/qgisplugins/qgisbostaskdepplugin/data</editform>
        <editforminit/>
        <editforminitcodesource>0</editforminitcodesource>
        <editforminitfilepath/>
        <editforminitcode></editforminitcode>
        <featformsuppress>0</featformsuppress>
        <editorlayout>generatedlayout</editorlayout>
        <editable/>
        <labelOnTop/>
        <widgets/>
        <previewExpression>"FID"</previewExpression>
        <mapTip/>
      </maplayer>
    </maplayers>
  </qlr>
```

26.3 Fig. : The top level tags of a QLR file

26.1.3 QML - The QGIS Style File Format

QML is an XML format for storing layer styling.

A QML file contains all the information QGIS can handle for the rendering of feature geometries including symbol definitions, sizes and rotations, labelling, opacity and blend mode and more.

The figure below shows the top level tags of a QML file (with only `renderer_v2` and its `symbol` tag expanded).

```
−<qgis version="3.4.13-Madeira" styleCategories="AllStyleCategories" readOnly="0" maxScale="0"
 labelsEnabled="0" simplifyDrawingHints="1" hasScaleBasedVisibilityFlag="0" simplifyDrawingTol="1"
 simplifyMaxScale="1" minScale="1e+8" simplifyAlgorithm="0" simplifyLocal="1">
 +<flags></flags>
 −<renderer-v2 symbollevels="0" enableorderby="0" type="singleSymbol" forceraster="0">
   −<symbols>
     +<symbol clip_to_extent="1" name="0" alpha="1" type="fill" force_rhr="0"></symbol>
    </symbols>
    <rotation/>
    <sizescale/>
  </renderer-v2>
 +<customproperties></customproperties>
  <blendMode>0</blendMode>
  <featureBlendMode>0</featureBlendMode>
  <layerOpacity>1</layerOpacity>
 +<SingleCategoryDiagramRenderer diagramType="Histogram" attributeLegend="1">
  </SingleCategoryDiagramRenderer>
 +<DiagramLayerSettings priority="0" linePlacementFlags="18" dist="0" showAll="1" placement="1"
  obstacle="0" zIndex="0">
  </DiagramLayerSettings>
 +<geometryOptions removeDuplicateNodes="0" geometryPrecision="0"></geometryOptions>
 +<fieldConfiguration></fieldConfiguration>
 +<aliases></aliases>
  <excludeAttributesWMS/>
  <excludeAttributesWFS/>
 +<defaults></defaults>
 +<constraints></constraints>
 +<constraintExpressions></constraintExpressions>
  <expressionfields/>
 +<attributeactions></attributeactions>
 +<attributetableconfig actionWidgetStyle="dropDown" sortExpression="" sortOrder="0">
  </attributetableconfig>
 +<conditionalstyles></conditionalstyles>
  <editform tolerant="1"/>
  <editforminit/>
  <editforminitcodesource>0</editforminitcodesource>
  <editforminitfilepath/>
 +<editforminitcode></editforminitcode>
  <featformsuppress>0</featformsuppress>
  <editorlayout>generatedlayout</editorlayout>
 +<editable></editable>
 +<labelOnTop></labelOnTop>
  <widgets/>
  <previewExpression>ID</previewExpression>
  <mapTip/>
  <layerGeometryType>2</layerGeometryType>
</qgis>
```

26.4 Fig. : The top level tags of a QML file (only the renderer_v2 tag with its symbol tag is expanded)

26.2 Appendix D: QGIS R script syntax

Contributed by Matteo Ghetta - funded by Scuola Superiore Sant'Anna

Writing R scripts in Processing is a bit tricky because of the special syntax.

A Processing R script starts with defining its **Inputs** and **Outputs**, each preceded with double hash characters (##).

Before the inputs, the group to place the algoritm in can be specified. If the group already exists, the algorithm will be added to it, if not, the group will be created. In the example below, the name of the group is *My group*:

```
##My Group=group
```

26.2.1 Inputs

All input data and parameters have to be specified. There are several types of inputs:

- vector: `##Layer = vector`
- vector field: `##F = Field Layer` (where *Layer* is the name of an input vector layer the field belongs to)
- raster: `##r = raster`
- table: `##t = table`
- number: `##Num = number`
- string: `##Str = string`
- boolean: `##Bol = boolean`
- elements in a dropdown menu. The items must be separated with semicolons `;`: `##type=selection point;lines;point+lines`

26.2.2 Išvestys

As for the inputs, each output has to be defined at the beginning of the script:

- vector: `##output= output vector`
- raster: `##output= output raster`
- table: `##output= output table`
- plots: `##output_plots_to_html` (##showplots in earlier versions)
- To show R output in the *Result Viewer*, put > in front of the command whose output you would like to show.

26.2.3 Syntax Summary for QGIS R scripts

A number of input and output parameter types are offered.

Input parameter types

Parametras	Syntax example	Returning objects
vector	Layer = vector	sf object (or SpatialDataFrame object, if ##load_vector_using_rgdal is specified)
vector point	Layer = vector point	sf object (or SpatialDataFrame object, if ##load_vector_using_rgdal is specified)
vector line	Layer = vector line	sf object (or SpatialDataFrame object, if ##load_vector_using_rgdal is specified)
vector polygon	Layer = vector polygon	sf object (or SpatialPolygonsDataFrame object, if ##load_vector_using_rgdal is used)
multiple vector	Layer = multiple vector	sf object (or SpatialDataFrame objects if ##load_vector_using_rgdal is specified)
lentelė	Layer = table	dataframe conversion from csv, default object of `read.csv` function
field	Field = Field Layer	name of the Field selected, e.g. `"Area"`
raster	Layer = raster	RasterBrick object, default object of `raster` package
multiple raster	Layer = multiple raster	RasterBrick objects, default object of `raster` package
number	N = number	integer or floating number chosen
string	S = string	string added in the box
longstring	LS = longstring	string added in the box, could be longer then the normal string
selection	S = selection first;second;third	string of the selected item chosen in the dropdown menu
crs	C = crs	string of the resulting CRS chosen, in the format: `"EPSG:4326"`
extent	E = extent	Extent object of the `raster` package, you can extract values as `E@xmin`
point	P = point	when clicked on the map, you have the coordinates of the point
file	F = file	path of the file chosen, e.g. „/home/matteo/file.txt“
folder	F = folder	path of the folder chosen, e.g. „/home/matteo/Downloads“

A parameter can be **OPTIONAL**, meaning that it can be ignored.

In order to set an input as optional, you add the string `optional` **before** the input, e.g:

```
##Layer = vector
##Field1 = Field Layer
##Field2 = optional Field Layer
```

Output parameter types

Parametras	Syntax example
vector	Output = output vector
raster	Output = output raster
lentelė	Output = output table
file	Output = output file

Pastaba

You can save plots as `png` from the *Processing Result Viewer*, or you can choose to save the plot directly from the algorithm interface.

Script body

The script body follows R syntax and the **Log** panel can help you if there is something wrong with your script.

Remember that you have to load all additional libraries in the script:

```
library(sp)
```

26.2.4 Pavyzdžiai

Example with vector output

Let's take an algorithm from the online collection that creates random points from the extent of an input layer:

```
##Point pattern analysis=group
##Layer=vector polygon
##Size=number 10
##Output=output vector
library(sp)
spatpoly = as(Layer, "Spatial")
pts=spsample(spatpoly,Size,type="random")
spdf=SpatialPointsDataFrame(pts, as.data.frame(pts))
Output=st_as_sf(spdf)
```

Explanation (per line in the script):

1. `Point pattern analysis` is the group of the algorithm
2. `Layer` is the input **vector** layer
3. `Size` is a **numerical** parameter with a default value of 10
4. `Output` is the **vector** layer that will be created by the algorithm
5. `library(sp)` loads the **sp** library
6. `spatpoly = as(Layer, "Spatial")` translate to an sp object
7. Call the `spsample` function of the `sp` library and run it using the input defined above (`Layer` and `Size`)
8. Create a *SpatialPointsDataFrame* object using the `SpatialPointsDataFrame` function
9. Create the output vector layer using the `st_as_sf` function

That's it! Just run the algorithm with a vector layer you have in the QGIS Legend, choose the number of random point. The resulting layer will be added to your map.

Example with raster output

The following script will perform basic ordinary kriging to create a raster map of interpolated values from a specified field of the input point vector layer by using the `autoKrige` function of the `automap` R package. It will first calculate the kriging model and then create a raster. The raster is created with the `raster` function of the raster R package:

```
##Basic statistics=group
##Layer=vector point
##Field=Field Layer
##Output=output raster
##load_vector_using_rgdal
require("automap")
require("sp")
require("raster")
```

(continues on next page)

(tęsinys iš praeito puslapio)

```
table=as.data.frame(Layer)
coordinates(table)= ~coords.x1+coords.x2
c = Layer[[Field]]
kriging_result = autoKrige(c~1, table)
prediction = raster(kriging_result$krige_output)
Output<-prediction
```

By using `##load_vector_using_rgdal`, the input vector layer will be made available as a `SpatialDataFrame` objects, so we avoid having to translate it from an `sf` object.

Example with table output

Let's edit the `Summary Statistics` algorithm so that the output is a table file (csv).

The script body is the following:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
##Stat=Output table
Summary_statistics<-data.frame(rbind(
    sum(Layer[[Field]]),
    length(Layer[[Field]]),
    length(unique(Layer[[Field]])),
    min(Layer[[Field]]),
    max(Layer[[Field]]),
    max(Layer[[Field]])-min(Layer[[Field]]),
    mean(Layer[[Field]]),
    median(Layer[[Field]]),
    sd(Layer[[Field]]))),
  row.names=c("Sum:","Count:","Unique values:","Minimum value:","Maximum value:",
↪"Range:","Mean value:","Median value:","Standard deviation:"))
colnames(Summary_statistics)<-c(Field)
Stat<-Summary_statistics
```

The third line specifies the **Vector Field** in input and the fourth line tells the algorithm that the output should be a table.

The last line will take the `Stat` object created in the script and convert it into a `csv` table.

Example with console output

We can use the previous example and instead of creating a table, print the result in the **Result Viewer**:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
Summary_statistics<-data.frame(rbind(
sum(Layer[[Field]]),
length(Layer[[Field]]),
length(unique(Layer[[Field]])),
min(Layer[[Field]]),
max(Layer[[Field]]),
max(Layer[[Field]])-min(Layer[[Field]]),
mean(Layer[[Field]]),
median(Layer[[Field]]),
sd(Layer[[Field]])),row.names=c("Sum:","Count:","Unique values:","Minimum value:",
↪"Maximum value:","Range:","Mean value:","Median value:","Standard deviation:"))
colnames(Summary_statistics)<-c(Field)
>Summary_statistics
```

The script is exactly the same as the one above except for two edits:

1. no output specified (the fourth line has been removed)
2. the last line begins with >, telling Processing to make the object available through the result viewer

Example with plot

To create plots, you have to use the `##output_plots_to_html` parameter as in the following script:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
##output_plots_to_html
####output_plots_to_html
qqnorm(Layer[[Field]])
qqline(Layer[[Field]])
```

The script uses a field (`Field`) of a vector layer (`Layer`) as input, and creates a *QQ Plot* (to test the normality of the distribution).

The plot is automatically added to the Processing *Result Viewer*.

26.3 Priedas E: QGIS aplikacijos tinklo jungtys

Tai sąrašas iš anksto sukonfigūruotų/automatinių tinklo jungčių, kurias kuria QGIS. Kai kurios jungtys inicijuojamos, nes iki jų įvykdymo reikia naudotojo veiksmo, kitos vyksta automatiškai.

Pavadini	Tikslas	UX režimas	Serveris	Į serverį siunčiama informacija	Serveryje saugoma informacija
qgis.org					
Python API pagalba	Naršyklės PyQGIS dokumentacija	Inicijuojama naudotojo	https://qgis.org/pyqgis/3.44/search	IP, QGIS versija, OS	IP serverio žurnale
version.q					
Naujos versijos tikrinima	Pranešti naudotojui, kad yra nauja QGIS versija	Automatinis	https://version.qgis.org	IP, QGIS versija, OS	IP serverio žurnale
feed.qgis					
QGIS naujienos	Gauti iš srauto QGIS naujienas	Automatinis	https://feed.qgis.org	IP, QGIS versija, kalbos kodas, paskutinio atsiuntimo laikas, OS	IP serverio žurnale, QGIS versija, OS ir IP agreguojami ir naudojami statistikos rinkimui
plugins.q					
Priedų atnaujinin tikrinima	Pranešti naudotojui apie priedų atnaujinimus	Inicijuojama naudotojo/auton (konfigūruojama	https://plugins.qgis.org	IP, QGIS versija, OS	IP serverio žurnale
Priedų sąrašas	Gauti priedų sąrašą	Inicijuojama naudotojo	https://plugins.qgis.org	IP, QGIS versija, OS	IP serverio žurnale
Priedų diegimas	Atsiųsti ir įdiegti priedo paketą	Inicijuojama naudotojo	https://plugins.qgis.org	IP, QGIS versija, OS	Priedo atsiuntimo skaitiklis padidinamas vienetu
Stiliai	Išvardinti naudotojų pateiktus stilius	Inicijuojama naudotojo	https://hub.qgis.org/styles	IP, QGIS versija, OS	Atsiuntimo skaitiklis padidinamas vienetu
3ių šalių					
Paviršiaus duomenys	Kurti DEM skirtą 3D vaizdui	Inicijuojama naudotojo	https://s3.amazonaws.com/elevation-tiles-prod/terrarium/{z}/{x}/{y	IP, QGIS versija, OS	žr. Amazon TOS
Google Map geokodav	Geokodavimo paslaugos	Inicijuojama naudotojo	https://maps.googleapis.com/maps/api/geocode/json	IP, QGIS versija, OS	Žr. googla maps API TOS
Nominati geokodav	Geokodavimo paslaugos	Inicijuojama naudotojo	https://nominatim.qgis.org/ui/search.html	IP, QGIS versija, OS	
Geodezin tinklelis	Pridėti naują PROJ tinklelį	Inicijuojama naudotojo	https://cdn.proj.org	IP, PROJ versija	Prieigos žurnalas visiškai ištrinamas po vienos dienos

SKYRIUS 27

Literature and Web References

GDAL-SOFTWARE-SUITE. Geospatial data abstraction library. https://gdal.org/en/latest, 2013.

GRASS-PROJECT. Geographic resource analysis support system. https://grass.osgeo.org, 2013.

NETELER, M., AND MITASOVA, H. Open source gis: A grass gis approach, 2008.

OGR-SOFTWARE-SUITE. Geospatial data abstraction library. https://gdal.org/en/latest, 2013.

OPEN-GEOSPATIAL-CONSORTIUM. Web map service (1.1.1) implementation specification. https://portal.ogc.org/files/?artifact_id=1081&version=1&format=pdf, 2002.

OPEN-GEOSPATIAL-CONSORTIUM. Web map service (1.3.0) implementation specification. https://portal.ogc.org/files/?artifact_id=14416&format=pdf, 2004.

POSTGIS-PROJECT. Spatial support for postgresql. http://www.refractions.net/products/postgis/, 2013.